
Universidade de Aveiro
2007

Departamento de Electrónica,
Telecomunicações e Informática

Roberto Nogueira **Simulação de sistemas P2P e estudo do seu tráfego característico**


Universidade de Aveiro
2007

Departamento de Electrónica,
Telecomunicações e Informática

Roberto Nogueira

Simulação de sistemas P2P e estudo do seu tráfego característico

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Doutor Paulo Salvador, Professor Auxiliar convidado e do Doutor António Nogueira, Professor Auxiliar, ambos do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

Dedico este trabalho ao meu falecido pai, meu guia e minha inspiração.

O júri

Presidente

Doutor Paulo Monteiro

Professor Associado da Universidade de Aveiro

Doutor Joel Rodrigues

Professor Auxiliar da Universidade da Beira Interior

Doutor António Nogueira

Professor Auxiliar da Universidade de Aveiro

Doutor Paulo Salvador

Professor Auxiliar da Universidade de Aveiro

Agradecimentos

Em primeiro lugar, gostaria de agradecer à minha esposa pelo incansável apoio, dedicação, e gestão da vida familiar.

Aos meus pais e irmã pelos incentivos nos momentos mais exigentes.

À minha entidade patronal por ter apostado na minha formação na medida em que para além de arcar com as despesas, mostrou total abertura à realização das actividades extra-laborais.

Um especial agradecimento ao Doutor António Nogueira e ao Doutor Paulo Salvador pela orientação do trabalho, a discussão de ideias e restantes sinergias.

Palavras-chave

Estudo de protocolos *P2P*, análise estatística, componentes principais, captura e caracterização de tráfego

Resumo

A Internet nasceu no final dos anos 60 como solução para as necessidades de incremento da capacidade computacional. Desde então, surgiu um vasto número de aplicações de comunicação e distribuição de dados, tais como a *World Wide Web* e as aplicações *Peer to Peer* (***P2P***), que contribuíram para a generalização da utilização da Internet, tornando-a no principal meio de divulgação/partilha de informação.

Na última década, as redes *P2P* e em particular os sistemas de partilha de ficheiros tiveram um crescimento exponencial na sua utilização, sendo responsáveis por cerca de 80% do volume de tráfego Internet actual.

Estes valores são incomportáveis para as actuais infra-estruturas de rede, inviabilizando qualquer garantia de qualidade de serviço por parte dos *ISP*s mundiais. Assim, afigura-se fundamental a análise e caracterização do tráfego gerado por aplicações *P2P*, como base ao desenvolvimento de ferramentas que limitem a largura de banda utilizada.

Neste documento encontra-se descrito o conjunto de actividades desenvolvidas com vista a esse objectivo:

- Estudo das aplicações *P2P* de partilha de ficheiros e respectivos protocolos;
- Identificação das aplicações Internet com maior utilização ou responsáveis por volume de tráfego elevado;
- Realização de ensaios de características semelhantes para a captura e análise de tráfego gerado por diversas aplicações: *bitTorrent*, *eMule*, *Shareaza*, *youTube*, *Skype*, etc.

A ferramenta ***TSTAT*** (*TCP Statistic and Analysis Tool*) permitiu a geração de dados estatísticos, com base nos resultados das capturas. Contudo, o elevado número de parâmetros presente nestes dados estatísticos invalida qualquer tentativa de análise gráfica no sentido da identificação de características próprias de cada aplicação.

A Análise de Componentes Principais (***PCA***) é uma técnica vulgarmente usada na compressão de dados, que permite a redução de dimensionalidade de um conjunto, mantendo as características do conjunto original.

Com base no resultado da Análise de Componentes Principais foi desenvolvido e testado um método de identificação de padrões de tráfego únicos.

Os resultados obtidos provam que o tráfego gerado por cada uma das aplicações *P2P* analisadas apresenta características únicas, que possibilitam a sua identificação no tráfego Internet global.

INDICE

FIGURAS

TABELAS

1 INTRODUÇÃO

Nos últimos anos assistiu-se a um aumento exagerado do tráfego Internet, quer em termos de volume quer na variedade de aplicações disponibilizadas. O aparecimento de aplicações em tempo-real de transmissão de voz e vídeo mudou de forma significativa a forma de utilização da Internet. Esta mudança contribuiu para a necessidade de remodelação do modo de tratamento do seu tráfego. A maioria destas aplicações baseia-se em protocolos *P2P*.

Neste capítulo são apresentados os principais objectivos da dissertação, bem como as actividades desenvolvidas para o cumprimento dos mesmos.

1.1 RESUMO DAS ACTIVIDADES

O aumento de utilizadores, associado ao constante aparecimento de novas aplicações Internet, contribuiram para um crescimento alarmante do tráfego gerado, que se transformou nesta última década na principal preocupação dos ISPs mundiais.

O documento apresenta um estudo dos serviços *P2P*, com particular atenção para a partilha de ficheiros. Para tal foi realizado o seguinte conjunto de actividades:

- Estudo dos principais protocolos *P2P* existentes.
- Captura e análise do tráfego gerado por cada aplicação.
- Aplicação de técnicas de identificação de padrões sobre o tráfego capturado.

1.1.1 ESTUDO DE PROTOCOLOS P2P

A primeira tarefa realizada consistiu na identificação das aplicações com maior difusão/geração de tráfego Internet. Dado a maioria serem aplicações *P2P* de partilha de ficheiros, elas constituem o principal foco da dissertação.

Para os ensaios realizados foram escolhidas não só aplicações que apresentam grandes comunidades de utilizadores mas também as que mais contribuem para o tráfego Internet global.

1.1.2 CAPTURA E ANÁLISE DE TRÁFEGO

Devido à natureza das aplicações e às complexas questões legais que as envolvem, tomou-se a opção de realizar as capturas de tráfego em ambiente laboratorial "caseiro". Os custos desta opção foram assumidos por mim: não só no que refere ao tráfego gerado, mas essencialmente no que toca ao material informático, ligação Internet e instalações necessárias ao correcto desenvolvimento dos trabalhos.

Para cada aplicação foi feito um conjunto de ensaios, com características semelhantes: duração, pesquisas realizadas, ficheiros partilhados/transferidos e configuração do par de rede observado. As mensagens trocadas entre a máquina usada nos testes e os restantes pares de rede foram capturadas através da aplicação ***WinDump*** ([8] e [9]). Os ficheiros resultantes da captura não oferecem por si só uma fonte de informação amigável para a análise de tráfego pretendida, atendendo à forma desagregada como a informação se apresenta. A ferramenta ***TSTAT*** ([6] e [7]) permite obter, com base no ficheiro *WinDump* da captura, um conjunto de estatísticas globais características do tráfego.

1.1.3 CARACTERIZAÇÃO DE TRÁFEGO CAPTURADO

Apesar das estatísticas geradas pelo *TSTAT* reduzirem significativamente a dimensionalidade da informação recolhida nas capturas, para a caracterização do tráfego é essencial reduzir e normalizar a informação permitindo assim a sua generalização.

Uma das técnicas mais difundidas e eficazes para a simplificação e interpretação de dados é a Análise de Componentes Principais.

Neste documento são descritos os procedimentos desenvolvidos, com o objectivo de automatizar o processo de identificação das componentes principais e caracterização do tráfego de cada aplicação.

1.1.4 ESTRUTURA DA DISSERTAÇÃO

Para além deste capítulo introdutório, a presente dissertação possui mais cinco capítulos relativos ao estudo, análise e desenvolvimento de técnicas de caracterização de aplicações *P2P*.

No **Capítulo 2**, PROTOCOLOS E APLICAÇÕES P2P, apresentam-se as principais aplicações *P2P* de partilha de ficheiros. Para além do enquadramento histórico é feita a

descrição do modo de operação protocolar de cada aplicação. Por último, é feita uma breve apresentação de cada uma das aplicações seleccionadas para os ensaios, bem como o motivo da sua escolha.

No **Capítulo 3**, AMBIENTE LABORATORIAL, é feita a descrição pormenorizada do *hardware* e *software* usado nos ensaios. De destacar a apresentação das aplicações que permitiram a captura de tráfego (*WinDump*) e posterior geração de dados estatísticos (*TSTAT*).

No **Capítulo 4**, ANÁLISE DE COMPONENTES PRINCIPAIS, encontra-se detalhado o método matemático *PCA* (*Principal Components Analysis*), assim como os conceitos matemáticos em que se sustenta.

O **Capítulo 5**, APRESENTAÇÃO DE RESULTADOS, encontra-se dividido em três secções:

- **1ª** Descrição dos ensaios realizados para as diversas aplicações;
- **2ª** Apresentação dos dados estatísticos recolhidos nas capturas de tráfego. Comparação dos valores agregados das aplicações e da representação bidimensional dos parâmetros considerados mais relevantes.
- **3ª** Análise das Componentes Principais dos dados estatísticos de cada aplicação. Desenvolvimento de técnicas para caracterização de tráfego de aplicações *P2P* de partilha de ficheiros.

No **Capítulo 6**, CONCLUSÕES, é feita uma retrospectiva do trabalho realizado e são apresentadas as principais conclusões, tentando perspectivar futuras evoluções.

2 PROTOCOLOS E APLICAÇÕES P2P

Geralmente uma rede *P2P* é constituída por computadores ou outros tipos de unidades de processamento que não possuem um papel fixo de cliente ou servidor, dependendo do início da transacção: ao par que inicia a transacção designa-se cliente; o par servidor é responsável pela resposta aos pedidos formulados pelo cliente, nessa transacção.

O termo *P2P* é utilizado em diferentes tecnologias que adoptam um modelo conceptual semelhante, tal como o protocolo ***NNTP*** (*Net News Transfer Protocol* [13], para *Usenet*), o protocolo ***SMTP*** (*Simple Mail Transfer Protocol* [14], para envio de *e-mail*) e diversos sistemas de troca de mensagens instantâneas (***Skype*** [15] e [16], *ICQ*, *MSN*, entre outros). Porém, o termo tornou-se popular com o aparecimento de aplicações de partilha de ficheiros em rede, permitindo o acesso de qualquer utilizador dessa rede aos conteúdos disponibilizados. Podem ainda ser partilhados outro tipo de recursos, tal como a capacidade de processamento de máquinas, espaço de armazenamento de ficheiros, entre outros.

2.1 PARTILHA DE FICHEIROS

Qualquer sistema *P2P* de partilha de ficheiros inclui os subsistemas de distribuição e pesquisa de ficheiros.

O subsistema de distribuição de ficheiros disponibiliza os meios de transmissão necessários entre os pares. O protocolo do sistema *P2P* define o comportamento dos pares no que diz respeito à transferência de ficheiros (*upload* ou *download*).

O subsistema de pesquisa de ficheiros fornece as ferramentas necessárias para que os utilizadores encontrem os ficheiros pretendidos, de entre os existentes na rede *P2P*. Este subsistema mantém tipicamente um índice actualizado dos ficheiros partilhados.

Como veremos ao longo deste capítulo, a implementação dos dois subsistemas difere de sistema *P2P* para sistema *P2P*.

2.1.1 NAPSTER

O ***Napster*** ([17] e [18]) é uma aplicação de partilha de ficheiros que permite a pesquisa e partilha de ficheiros MP3 entre utilizadores, através da Internet. Foi concebido por *Shawn Fanning* que, para além de desenvolver a aplicação, foi pioneiro no desenho de um protocolo que permite a comunicação directa entre pares clientes. Este protocolo serviu de base a outros grupos e organizações para o desenvolvimento de protocolos *P2P* mais complexos e eficientes.

2.1.1.1 ARQUITECTURA

A arquitectura do *Napster* baseia-se no modelo centralizado. Existe um sistema servidor central, responsável pela gestão do tráfego individual de cada utilizador registado. É também da sua responsabilidade a manutenção de um índice actualizado sobre todos os ficheiros MP3 partilhados pelos utilizadores. A actualização do índice é feita sempre que um utilizador se liga/desliga do servidor.

Sempre que um utilizador formula um pedido/pesquisa de um ficheiro, o servidor central cria uma lista de ficheiros que se enquadram no pedido, a partir da informação existente na sua base de dados. Essa lista é composta por um conjunto de links HTTP, que podem ser seleccionados pelo utilizador, estabelecendo uma ligação directa com um computador que possui o ficheiro pretendido.

O processo de transferência do ficheiro MP3 não tem intervenção do servidor central ou de qualquer outro intermediário. O ficheiro é trocado directamente entre o par que partilha e o par que formulou o pedido de transferência.

No que toca à partilha de ficheiros, o *Napster* é um sistema *P2P*. Contudo, a indexação de ficheiros apresenta claramente uma arquitectura centralizada.

2.1.2 GNUTELLA

Em Março de 2000, numa altura em que o *Napster* estava a ser investigado pela distribuição ilegal de material protegido por direitos de autor, *Justin Frankel* e *Tom Pepper* lançaram o ***Gnutella*** [19] no *website* Internet da *Gnullsoft* (subsidiária da *AOL*).

Passado pouco tempo, foram retirados os links de *download* da aplicação do *website* da *Gnullsoft* por imposição da *AOL*, devido à sua fusão com a *Time Warner Music*.

Os problemas do *Napster* com a justiça, aliados à natureza do *Gnutella* (*P2P* puro) fomentaram a rápida difusão da aplicação entre grupos de programadores que, através de *reverse-engineering*, especificaram o protocolo.

A divulgação do protocolo contribuiu para o rápido desenvolvimento/melhoramento das aplicações nele baseadas (por exemplo, *LimeWire*, *BearShare*, *Gnucleus*, *XoloX* e ***Shareaza*** [28]).

2.1.2.1 ARQUITECTURA

Ao contrário do *Napster* que recorre a uma directoria de indexação, o *Gnutella* tem uma rede de máquinas (pares) que funcionam simultaneamente como servidores e clientes e que mantêm a directoria de indexação dos conteúdos existentes no sistema.

Para que uma máquina se junte à rede *Gnutella* é necessário que conheça o endereço IP de uma máquina ligada à rede. Para este efeito podem ser usadas *caches* de máquinas que se encontram permanentemente ligadas à rede *Gnutella*.

Após a identificação de um endereço IP válido, a máquina que pretende ligar-se envia um pedido do tipo **GNUTELLA CONNECT**. Ao receber o pedido, a máquina que se encontra ligada à rede responde com uma mensagem do tipo **GNUTELLA OK**, ou rejeita-o enviando qualquer outra mensagem de resposta. A rejeição pode dever-se a várias razões: número máximo de ligações atingido, versões diferentes do protocolo, etc.

Uma vez ligada à rede, a máquina envia periodicamente mensagens do tipo **Ping** para as máquinas vizinhas com o objectivo de descobrir outras máquinas pertencentes à rede. Tipicamente, cada máquina deverá ligar-se a várias máquinas pertencentes à rede, dado o dinamismo da mesma (uma máquina pode ligar-se ou desligar-se da rede em qualquer momento). É por isso importante manter o contacto com várias máquinas ao mesmo tempo, de forma a prevenir uma eventual perda de ligação.

Quando uma máquina recebe uma mensagem do tipo **Ping**, responde com uma mensagem do tipo **Pong**, usando o mesmo percurso (em sentido inverso). Uma mensagem **Pong** contém informação relativa a máquinas pertencentes à rede: endereço IP, porto, número de ficheiros partilhados e o número de *kilobytes* partilhados.

No protocolo *Gnutella* cada máquina é responsável pela manutenção da sua directoria de indexação.

Para a procura de um ficheiro, a máquina envia uma mensagem do tipo **Query** para todas as máquinas vizinhas que conhece. Quando uma destas máquinas recebe o pedido, verifica internamente se o pode satisfazer, caso contrário encaminha-o para as máquinas vizinhas.

Quando uma máquina verifica que pode satisfazer o pedido, responde com uma mensagem do tipo **QueryHit**, pelo percurso feito pela **Query**. Contudo se esta máquina se encontra ligada através de uma *Firewall*, não é possível à máquina que realizou o pedido estabelecer a ligação para a transferência do conteúdo. Neste caso, a máquina que formulou o pedido envia uma mensagem do tipo **Push**, para iniciar a ligação para a transferência.

De notar que a transferência de conteúdos não é feita através do protocolo G*nutella*, mas sim por *HTTP*.

2.1.3 FASTTRACK

O protocolo ***FastTrack*** [20] é propriedade da empresa *Sherman Networks*. Por esse motivo, é um protocolo com especificação fechada, sobre o qual existe pouca informação. No entanto existiram várias tentativas para o decifrar, de entre as quais se destaca o projecto ***giFT*** [21] que, apesar de ter esbarrado na encriptação do protocolo, permitiu compreender melhor o seu modo de funcionamento.

2.1.3.1 ARQUITECTURA

Esta tecnologia recorre a dois níveis de controlo. O primeiro apresenta uma topologia centralizada: existe uma máquina central com ligação de banda larga, designada de super-nó, à qual se encontra ligado um conjunto de máquinas de utilizadores comuns.

O segundo nível é formado pelos super-nós existentes na rede, que se encontram ligados de forma descentralizada.

O número de super-nós numa rede *FastTrack* é muito volátil (das dezenas aos milhares), dado tratarem-se de máquinas "comuns" que se ligam e desligam da rede frequentemente.

Para garantir a constante disponibilidade da rede, foi necessária a introdução de um novo elemento na arquitectura: nó de *bootstrapping*, responsável pela monitorização e controlo das ligações. Estas máquinas encontram-se sempre activas na rede.

Quando uma aplicação *FastTrack* arranca, envia um pedido de ligação à rede para nó de *bootstrapping*.

A informação presente no pedido permite ao nó de *bootstrapping* determinar se a máquina do cliente é ou não um super-nó. Em caso afirmativo, envia na resposta os endereços IP dos super-nós activos na rede. Caso contrário, indica o endereço IP de um dos super-nós activos, ao qual o cliente poderá ligar-se.

Algumas das aplicações *FastTrack*, como o ***Kazaa*** [22], utilizam um método designado como "sistema de reputação". A reputação de um dado utilizador reflecte-se no seu nível de participação na rede (um número entre 0 e 1000). Quanto mais tempo o utilizador estiver ligado à rede, maior será o seu nível de participação, sendo favorecido nas políticas de filas de espera implementadas no protocolo e consequentemente no serviço recebido. Este método pretende encorajar os utilizadores a partilharem ficheiros e reduzir de modo eficiente os chamados "clientes passageiros".

Os super-nós são responsáveis pela resposta aos pedidos de conteúdos na rede. Quando uma máquina do primeiro nível realiza uma pesquisa, envia o pedido para o super-nó a que se encontra ligada. Por sua vez, este espalha o pedido pelos restantes super-nós. Este processo é repetido até que o parâmetro *TTL* atinja zero.

Suponhamos um *TTL* de 7 e que existem em média 10 máquinas ligadas a cada super-nó. Nestas condições, um cliente *FastTrack* tem acesso à informação de 11 máquinas, mais do que teria numa configuração semelhante na rede *Gnutella*.

A arquitectura *FastTrack* favorece os seus clientes, oferecendo uma maior cobertura e melhores resultados nas pesquisas realizadas. A difusão dos pedidos dos clientes entre os super-nós sobrecarrega o processamento destas máquinas e cria um elevado volume de tráfego. Este problema é semelhante ao existente no mecanismo de troca de mensagens da rede *Gnutella*. Contudo, no caso do *FastTrack* esta questão é atenuada devido à banda larga da ligação entre os super-nós.

Actualmente o protocolo *FastTrack* tem pouca relevância no tráfego global Internet, devido ao facto da aplicação mais conhecida, o *Kazaa*, ter passado a ser paga. Por este motivo o protocolo *FastTrack* não foi analisado, apesar do enorme relevo histórico que teve no desenvolvimento das aplicações *P2P* de partilha de ficheiros.

2.1.4 EMULE

A rede ***eDonkey*** [23], criada em 2000 pela empresa alemã *MetaMachine*, tornou-se em 2004 a rede de partilha de ficheiros com maior número de utilizadores, destronando a até então líder *FastTrack*.

A rede *eDonkey* original era suportada por um conjunto de servidores centrais que garantiam a largura de banda, disco e processamento necessários. Devido a esta configuração da rede, os servidores estavam sujeitos a elevadas cargas de tráfego, tornando-os vulneráveis a ataques.

Para ultrapassar este problema, a *MetaMachine* criou o ***Overnet*** [23] com o intuito de suceder ao protocolo *eDonkey*. Devido à sua topologia descentralizada (não tem servidores), o tráfego é mais distribuído pela rede, tornando-o muito mais rápido.

Os criadores do ***eMule*** [23] desenvolveram uma nova rede de características idênticas à *Overnet* a que deram o nome de ***Kademlia*** [23], suportada nas versões mais recentes do software *eMule*.

2.1.4.1 ARQUITECTURA

A rede *eMule* é composta por várias centenas de servidores e milhões de clientes.

Para que um cliente se junte à rede, é necessário que envie um pedido de ligação para um dos servidores activos. Esta ligação mantém-se activa enquanto o cliente se encontrar no sistema.

Os servidores são responsáveis pelos serviços de indexação (tal como acontecia no *Napster*), não existindo qualquer tipo de transferência de informação entre servidores.

Cada cliente *eMule* é pré-configurado com uma lista de servidores que pode ser actualizada através de listas existentes na Internet. Para se ligar à rede, o cliente abre uma ligação TCP para um dos servidores existentes, recolhendo informação relativa a ficheiros que procura e clientes que os partilham. O servidor atribui um identificador ao cliente (*clientID*), usado para identificá-lo na rede durante a sessão que iniciou. Não é possível a um cliente ligar-se a mais do que um servidor simultaneamente, nem tão pouco mudar dinamicamente de servidor, sem intervenção do utilizador.

Após o estabelecimento da ligação, é trocado um elevado número de mensagens entre o cliente e o servidor. Estas mensagens têm por objectivo actualizar a informação que ambos mantêm.

A pesquisa de ficheiros é iniciada pelo utilizador. A operação é simples: o cliente envia um pedido de pesquisa, através de uma mensagem do tipo **Search Request**; por seu turno o servidor responde com uma mensagem do tipo **Search Result**, contendo o resultado da pesquisa. Posteriormente, o cliente pede fontes para os ficheiros consoante a escolha do utilizador. A este pedido, o servidor envia uma lista de fontes disponíveis. Usualmente, antes desta lista o servidor envia uma mensagem do tipo **Server Status** com informação relativa ao número de utilizadores e ficheiros suportados actualmente pelo servidor.

Após verificar que existem fontes novas nos resultados, o cliente tenta ligar-se a cada uma delas e adiciona as válidas à sua lista. A ordem definida na lista de resultados define a ordem pela qual as fontes são contactadas pelo cliente.

O mecanismo de transferência de ficheiros está desenhado de forma a ultrapassar a falta de capacidade por parte dos clientes com *ClientID* baixo para aceitar pedidos doutros clientes, ou seja dificuldades na partilha de ficheiros.

2.1.5 BITTORRENT

Em 2001 *Bram Cohen* criou o protocolo ***BitTorrent*** e a aplicação com o mesmo nome que o implementa ([24] e [25]). Actualmente é o protocolo *P2P* de partilha de ficheiros mais popular, representando cerca de 53% do tráfego gerado na Internet por aplicações *P2P*, segundo um estudo revelado pela *CNN*.

Ao contrário dos sistemas *P2P* tradicionais (por exemplo *Gnutella*, *FastTrack* e *eMule*), o *BitTorrent* organiza os pares da sua rede segundo o tipo de ficheiros que partilham, com o objectivo da distribuição rápida e eficiente de cada ficheiro.

2.1.5.1 ARQUITECTURA

O *BitTorrent* foi o primeiro sistema *P2P* de partilha de ficheiros a desagregar o subsistema de distribuição de ficheiros do subsistema de pesquisa de ficheiros.

A distribuição de ficheiros é assegurada pelos pares que compõe a rede *BitTorrent*.

O subsistema de pesquisa de ficheiros é suportado por um conjunto de *websites* que garantem um índice actualizado dos ficheiros partilhados na rede *BitTorrent*, disponibilizando a interface para as pesquisas dos utilizadores. O índice referido é mantido por máquinas que não são pares *BitTorrent*, pelo que se pode considerar um sistema com indexação centralizada.

Para além dos *websites*, existem outras aplicações Internet onde é possível encontrar ficheiros na rede *BitTorrent*, tais como o e-mail e o *IRC* (*Internet Relay Chat*).

Existe um conjunto de novidades nos protocolos *P2P* que foram introduzidas pelo sistema de partilha de ficheiros *BitTorrent*. Uma das principais novidades encontra-se no protocolo de distribuição de ficheiros, que obriga à divisão dos ficheiros em fragmentos. Assim, durante a transferência integral de um ficheiro, o par cliente é obrigado a partilhar os fragmentos entretanto transferidos com os restantes pares da rede.

Outra das novidades é a política de gratificações do subsistema de distribuição de ficheiros, que compensa os pares que partilham ficheiros, atribuindo-lhes maior velocidade nas transferências por eles iniciadas.

Um utilizador que pretenda partilhar um ou mais ficheiros, deverá usar um programa de criação de ficheiros *torrent*.

Um *torrent* contém o nome do ficheiro, o resultado *hash* **SHA-1** [26] de cada fragmento do ficheiro, o *infohash* e o endereço *web* de um ou mais *trackers* que pode ser usado. O *infohash* é o resultado *hash* SHA-1 de todos os fragmentos do ficheiro, sendo usado para identificar univocamente o ficheiro *torrent* no sistema *BitTorrent*. O *tracker* é um servidor que mantém uma lista de endereços IP de pares dentro do *swarm*. Dá-se o nome de *swarm* ao conjunto de pares que, num determinado momento, se encontram envolvidos na transmissão do ficheiro, de uns para os outros.

Para partilhar o ficheiro *torrent*, existem meios externos ao par *BitTorrent* (por exemplo, *websites*) que permitem a sua divulgação.

Na terminologia *BitTorrent*, designa-se por *seed* ao par que partilha o ficheiro completo. Quando um utilizador publica um ficheiro *torrent*, o seu par é o único *seed* do *swarm*. Sempre que um par termina a transferência do ficheiro, torna-se *seed* desse ficheiro.

Um utilizador interessado na transferência de um determinado ficheiro do sistema deverá obter, em primeiro lugar, o ficheiro *torrent* correspondente. Após o *download* do

torrent, a aplicação *BitTorrent* do utilizador inicia o processo de transferência do ficheiro. O par liga-se ao *tracker* para obter um conjunto de endereços IP de pares remotos, pertencentes ao *swarm*. Os endereços IP retornados representam um subconjunto aleatório da lista completa mantida pelo *tracker*.

O pedido formulado pelo par permite ao *tracker* a actualização da lista do *swarm* do ficheiro, com os dados deste novo par. Com a informação presente na resposta, o par tenta ligar-se de forma aleatória a um determinado número de pares que partilham o ficheiro. Usualmente este número encontra-se entre os 4 e 30 pares, dependendo da configuração e aplicação do utilizador.

Após facultar o endereço IP ao *tracker*, o par deverá começar a receber pedidos de ligação de outros pares que pretendam a transferência do ficheiro. Quando um par se liga a um par remoto, trocam entre si um mapa de bits com informação relativa aos fragmentos completos do ficheiro que cada um deles tem. Esta interacção permite verificar se o par remoto partilha fragmentos que ainda não possui.

O resultado deste comportamento em cada par é o seguinte: inicialmente o par que publicou o *torrent* é o único *seed*, pelo que é contactado pelos pares remotos do *swarm* para a transferência do ficheiro. Imediatamente após um par remoto completar a transferência de um fragmento do ficheiro, inicia a sua transmissão no *swarm*, enquanto recebe os fragmentos do ficheiro que faltam.

Este comportamento permite a rápida propagação dos fragmentos no *swarm*. De notar que o *seed* inicial só precisa de transmitir cada fragmento do ficheiro uma vez, para que seja possível a total distribuição do ficheiro pelo *swarm*, independentemente do número de pares que o constitua. Este cenário demonstra as potencialidades de escalabilidade da arquitectura *BitTorrent*.

Um dos maiores problemas enfrentados pelos sistemas *P2P* de partilha de ficheiros é colocado pelos designados *free-riders*, utilizadores que após o *download* do ficheiro não o partilham. A existência deste tipo de utilizadores contribui para baixas velocidades ou impossibilidade de transferência, devido à escassez de fontes.

O *BitTorrent* implementa um esquema de justiça que encoraja os pares a partilhar os fragmentos/ficheiros transferidos. Este esquema define que o par só disponibiliza o

ficheiro a um subconjunto de pares remotos a que se encontra ligado – os pares remotos que mais contribuem para a transferência actual do par.

2.2 APLICAÇÕES SELECCIONADAS

O artigo "***P2P: The Next Wave Of Internet Evolution***" de *John G. Waclawsky* [27] serviu de base à selecção das aplicações *P2P* usadas nos ensaios. As estatísticas do *website download.com* permitiram refinar esta selecção.

Todas as aplicações são gratuitas (*freeware*) e encontram-se disponíveis no *website download.com*.

2.2.1 SHAREAZA 2.2.5

O *Shareaza* é uma aplicação baseada no protocolo *Gnutella* (descrito no ponto 2.1.2), desenvolvida através do projecto *sourceforge.net*. Trata-se de um projecto aberto a qualquer utilizador Internet, que permite o desenvolvimento da aplicação por múltiplos utilizadores. O acesso ao código fonte e o controlo das alterações realizadas é gerido pelo ***CVS*** (*Concurrent Versions System*, [29]) do projecto.

Aquando da transferência da aplicação, o *website download.com* registava cerca de 3 milhões de transferências.

Apesar da aplicação permitir a ligação a várias redes *Gnutella*, *Gnutella2*, *eMule* e *bitTorrent*, apenas foi usada para análise do protocolo *Gnutella*.

2.2.2 EMULE 0.48A

A insatisfação de *Hendrik Breitkreuz* (também conhecido por *Merkur*) relativamente às funcionalidades/interface oferecido pela aplicação *eDonkey2000* levou-o a criar o *eMule*. Posteriormente a aplicação foi tornada pública no projecto *sourceforge.net*. Actualmente é a aplicação com maior número de downloads do projecto. O protocolo que a aplicação implementa é descrito no ponto 2.1.4.

Aquando da transferência da aplicação, o *website download.com* registava cerca de 6 milhões de transferências.

2.2.3 BITTORRENT 5.0.5

A aplicação *bitTorrent* que *Bram Cohen* desenvolveu, implementa o protocolo de mesmo nome descrito no ponto 2.1.5. Entretanto foi criada a empresa *bitTorrent* que suporta as versões mais recentes da aplicação.

Aquando da transferência da aplicação, o *website download.com* registava cerca de 12 milhões de transferências.

2.2.4 SKYPE 3.5.0

A aplicação *Skype* é actualmente a aplicação *P2P* mais popular, no que diz respeito a telefonia por Internet (*VOIP*) e *instant messaging*.

Ao contrário dos sistemas não-*P2P*, onde todos os pacotes trocados entre clientes são encaminhados pelo servidor central, o protocolo implementado pelo *Skype* define a troca directa dos pacotes entre os pares. Nos casos em que essa troca directa não seja possível, ela será feita à custa do encaminhamento através de outros pares da rede *Skype*.

A infra-estrutura descentralizada da rede permite a sua escalabilidade sem custos. Actualmente, em qualquer momento estão ligados à rede cerca de 2 milhões de utilizadores simultaneamente.

Aquando da transferência da aplicação, o *website download.com* registava aproximadamente 3 milhões de transferências.

2.2.5 YOUTUBE

O ***youTube*** ([30] e [31]) é um serviço centralizado de partilha de vídeos, que permite o upload de ficheiros dos utilizadores para os seus servidores. Os ficheiros são publicados no *website www.youtube.com*, após a verificação da sua legalidade e do seu conteúdo.

Desde o lançamento em 2005, o *youTube* tornou-se um serviço extremamente popular. Actualmente são visionados mais de 100 milhões de vídeos por dia.

Para muitos utilizadores o *youTube* é visto como um veículo para iniciar uma carreira de produtor, realizador ou mesmo jornalista. Como exemplo, tome-se o *Manchester United* que contratou um rapaz de 10 anos após o visionamento de um vídeo publicado no *youTube* onde demonstrava algumas das suas qualidades futebolísticas.

3 AMBIENTE LABORATORIAL

Durante o planeamento e execução dos ensaios, foram tidos como essenciais as seguintes directivas:

- Preservação das configurações de *hardware* e *software*;
- Utilização das configurações *default* das aplicações, sempre que possível;
- Utilização de software *freeware* ou *shareware*.

3.1 HARDWARE

Os ensaios relativos a cada uma das aplicações anteriormente mencionadas foram realizados num computador portátil ***HP OmniBook XE3***, com as seguintes características:

- Processador: *Intel Celeron* 800 MHz;
- Memória: 256MB *SDRAM*;
- Disco Rígido: 20GB;
- LAN Ethernet 10/100 Mbps.

A ligação de 4 Mbps à rede *ADSL* da PT Comunicações foi assegurada através de um *Modem Router* ***BELKIN G+ MIMO***.

Durante o processo houve a preocupação de não alterar as condições de *hardware*, de forma a manter o cenário o mais estável possível.

3.2 SOFTWARE

Foi instalado o sistema operativo ***Microsoft Windows XP Version 2002*** com ***Service Pack 2*** no computador. A escolha teve em consideração a sua popularidade e o facto de ser a plataforma para a qual são desenvolvidas mais aplicações.

De forma a minorar o impacto nos ensaios, foram realizadas previamente as seguintes operações:

- *Download* das aplicações necessárias aos ensaios: *Shareaza*, *eMule*, *bitTorrent*, *Skype*, *WinPcap* e *WinDump*;
- Desactivação de *firewalls* do sistema operativo e *router*;
- Configuração das aplicações com parametrizações similares:
    - Limite máximo de *download*: 96 Kbps;
    - Limite máximo de *upload*: 10 Kbps;
    - Número máximo de ligações: 100.

3.2.1 WINDUMP

O ***WinDump*** ([8] e [9]) é a versão equivalente à ferramenta *TcpDump*, desenvolvida para *UNIX* por *Van Jacobson*, *Craig Leres* e *Steven McCanne*, para plataformas *Win32*. Fornece um interface por linha de comandos que permite armazenar em ficheiro o resultado da captura e filtragem do tráfego de rede gerado. Suporta os protocolos *IP*, *ICMP*, *ARP*, *RARP*, *UDP* e *TCP*.

Um exemplo típico de invocação da ferramenta é

windump –i 2 –C 700 –w "captureTest.dmp"

que captura todos os pacotes enviados e recebidos pela máquina no interface de rede 2 (-i 2), para o ficheiro captureTest.dmp (–w). A instrução –C 700, define um tamanho máximo em *MBytes* para o ficheiro gerado. Quando este limite é atingido, a aplicação passa a despejar a informação da captura de tráfego num novo ficheiro com o nome captureTest1.dmp. Para cada novo ficheiro criado, o valor da sequência é incrementado (o ficheiro seguinte será captureTest2.dmp).

O *website www.winpcap.org* disponibiliza a ferramenta *WinDump* para *download*, assim como a lista completa de opções e comandos existentes.

3.2.2 TSTAT

É gerada uma linha no ficheiro de despejo do *WinDump* para cada pacote enviado/recebido pela máquina. Para grandes volumes de tráfego, como é o caso, os ficheiros serão gigantescos, com vários milhões de linhas.

Com o objectivo de reduzir a complexidade da análise deste tipo de informação foi desenvolvida a aplicação ***TSTAT*** ([6] e [7]) que correlaciona os fluxos de entrada/saída da máquina com o objectivo de gerar estatísticas.

Actualmente, a aplicação só se encontra disponível para ambientes *Linux*. A tabela seguinte apresenta uma breve descrição dos campos presentes nos ficheiros gerados pela aplicação *TSTAT*.

	CAMPO	DESCRIÇÃO
CLIENTE / SERVIDOR	IP ADDRESS	Endereço IP
	TCP PORT	Porto TCP usado no fluxo
	PACKETS	Total de pacotes enviados
	RST SENT	1 – foram enviadas mensagens do tipo RST pela máquina; 0 – caso contrário. A mensagem do tipo RST é usada para fechar ou reiniciar uma ligação.
	ACK SENT	Total de mensagens do tipo ACK enviadas pela máquina. A mensagem do tipo ACK é enviada para confirmação da recepção de um segmento.
	PURE ACK SENT	Total de mensagens do tipo ACK sem *payload*, enviadas pela máquina.
	UNIQUE BYTES	Total de *Bytes* enviados no *payload* das mensagens.
	DATA PACKETS	Total de mensagens enviadas com *payload*.
	DATA BYTES	Total de *Bytes* enviados no *payload* das mensagens, incluindo retransmissões.
	REXMIT PACKETS	Total de mensagens retransmitidas.
	REXMIT BYTES	Total de *Bytes* retransmitidos.
	OUT SEQ PACKETS	Total de mensagens recebidas fora de ordem.
	SYN COUNT	Total de mensagens do tipo SYN enviadas pela máquina. A mensagem do tipo SYN é enviada para estabelecimento da ligação e definição da sequência de sincronização entre as máquinas (cliente/servidor).
	FIN COUNT	Total de mensagens do tipo FIN enviadas pela máquina. A mensagem do tipo FIN é usada para indicar o término do envio uma sequência de mensagens.
	RFC 1323 WS	1 – foi negociado o tamanho da janela; 0 – caso contrário.
	RFC 1323 TS	1 – mensagens com registo temporal; 0 – caso contrário.
	WINDOW SCALE	Tamanho da janela negociado (factor)
	SACK REQ	1 – máquina pediu ACKs selectivos; 0 – caso contrário.
	SACK SENT	Total de mensagens do tipo SACK enviadas pela máquina. A mensagem do tipo SACK é uma evolução da mensagem ACK, apresentando melhoramentos na gestão da sequência das mensagens.
	MSS	Tamanho máximo das mensagens declarado, em *Bytes*.
	MAX SEG SIZE	Tamanho máximo das mensagens observado, em *Bytes*.
	MIN SEG SIZE	Tamanho mínimo das mensagens observado, em *Bytes*.
	WIN MAX	Tamanho máximo da janela de recepção declarado nas mensagens, em *Bytes*.
	WIN MIN	Tamanho mínimo da janela de recepção declarado nas mensagens, em *Bytes*.
	WIN ZERO	Total de mensagens com janela de recepção com factor 0.

GENÉRICO	CWIN MAX	Tamanho máximo da janela de congestão observado, em *Bytes*.
	CWIN MIN	Tamanho mínimo da janela de congestão observado, em *Bytes*.
	INITIAL CWIN	Total de *Bytes* enviados antes da recepção da primeira mensagem do tipo ACK.
	AVERAGE RTT	Tempo médio de *Round Trip* (milissegundos). Tempo que medeia o envio de uma mensagem e a recepção da mensagem do tipo ACK correspondente.
	RTT MIN	Tempo mínimo entre o envio de uma mensagem e recepção da mensagem do tipo ACK correspondente (milissegundos).
	RTT MAX	Tempo máximo entre o envio de uma mensagem e recepção da mensagem do tipo ACK correspondente (milissegundos).
	STDEV RTT	Desvio padrão do Tempo de *Round Trip*, em milissegundos.
	RTT COUNT	Tempo de *Round Trip* válido observado, em milissegundos.
	TTL MIN	TTL mínimo observado.
	TTL MAX	TTL máximo observado.
	RTX RTO	Total de mensagens retransmitidas devido a tempo de expiração esgotado.
	RTX FR	Total de mensagens retransmitidas devido a *Fast Retransmit*.
	REORDERING	Total de mensagens de reordenação da sequência.
	NET DUP	Total de mensagens duplicadas observados na rede.
	UNKNOWN	Número de mensagens fora de sequência ou duplicadas, sem classificação.
	FLOW CONTROL	Total de mensagens de teste da janela de recepção.
	UNNECE RTX RTO	Total de mensagens transmitidas devido a tempo de expiração esgotado, desnecessariamente.
	UNNECE RTX FR	Total de mensagens transmitidas devido a *Fast Retransmit*, desnecessariamente.
	!= SYN SEQNO	1 – Mensagens SYN retransmitidas com número de sequência diferentes; 0 – caso contrário.
	COMPLETION TIME	Duração do fluxo, em milissegundos.
	FIRST TIME	Primeira mensagem trocada após a abertura da ligação (milissengundos).
	LAST TIME	Última mensagem trocada após a abertura da ligação (milissengundos).
	C FIRST PAYLOAD	Primeira mensagem com *payload* enviada pelo cliente, após a abertura da ligação (milissengundos).
	S FIRST PAYLOAD	Primeira mensagem com *payload* enviada pelo servidor, após a abertura da ligação (milissengundos).
	C LAST PAYLOAD	Última mensagem com *payload* enviada pelo cliente, após a abertura da ligação (milissengundos).
	S LAST PAYLOAD	Última mensagem com *payload* enviada pelo servidor, após a abertura da ligação (milissengundos).
	INTERNAL	1 – Cliente tem IP interno; 0 – caso contrário.

Tabela 1: Campos do ficheiro de saída TSTAT

4 ANÁLISE DE COMPONENTES PRINCIPAIS

Uma das dificuldades associadas a conjuntos de estatísticas multi-variados é o problema na visualização multidimensional. Não existe representação gráfica para conjuntos de dados com mais de 3 variáveis.

Felizmente quanto maior for o número de variáveis maior será a probabilidade da existência de variáveis correlacionadas que descrevem um mesmo comportamento do sistema. Um exemplo é o número de pacotes e de *KBytes*, que se referem ambos ao volume do tráfego gerado.

O método ***PCA*** (*Principal Components Analisys* [10]) é um método quantitativo rigoroso que se baseia neste conceito para a simplificação/diminuição do número de variáveis. Da aplicação do método resulta um sub-conjunto de variáveis designado por componentes principais.

4.1 BASE MATEMÁTICA

De seguida apresentam-se os principais conceitos matemáticos em que o método se baseia.

DESVIO PADRÃO

O desvio padrão de um conjunto de dados é uma medida do espalhamento dos seus pontos. Por definição representa a distância média entre um ponto e a média do conjunto.

$$s = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}}$$

VARIÂNCIA

A variância é outra medida do espalhamento de um conjunto de dados. Na prática, é a designação para o valor do desvio padrão ao quadrado.

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}$$

COVARIÂNCIA

A expressão apresentada para a variância, pode ser escrita da seguinte forma:

$$var(X) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})}{(n-1)}$$

A covarância é uma medida de relacionamento bidimensional baseada na fórmula anterior.

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

Um valor positivo da covariância indica que as dimensões crescem a par. Quando este valor é negativo as dimensões crescem em direcções opostas: quando uma cresce a outra decresce.

MATRIZ DE COVARIÂNCIA

Para conjuntos com mais de duas dimensões é possível calcular vários valores de variância, por combinação das dimensões. Por exemplo, para um conjunto de três dimensões (*x*, *y*, *z*) existem as covariâncias: *cov*(*x*, *y*), *cov*(*x*, *z*) e *cov*(*y*, *z*). Generalizando, é possível calcular $\frac{n!}{2(n-2)!}$ valores diferentes de covariância, onde n representa o número de dimensões.

A matriz formada pelos valores das covariâncias das diferentes dimensões de um conjunto fornece uma medida de relacionamento para conjuntos multidimensionais:

$$C^{m\times n} = (c_{i,j},\ c_{i,j} = cov(Dim_i, Dim_j))$$

Para conjuntos tridimensionais (*x*, *y*, *z*) a matriz assume o seguinte formato:

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

Podemos verificar que a diagonal da matriz é composta pela variância de cada dimensão. Para além disso, a matriz é simétrica dado que *cov*(*x*, *y*) = *cov*(*y*, *x*).

VALORES E VECTORES PRÓPRIOS

Genericamente, designa-se **vector próprio** de M a um vector não nulo $\vec{VP}$, se existe um escalar λ tal que $M\left(\vec{VP}\right) = \lambda\vec{VP}$. A λ chama-se **valor próprio** de M associado ao vector próprio $\vec{VP}$.

No cálculo matricial só é possível a determinação de valores e vectores próprios de algumas matrizes quadradas. Para uma matriz quadrada *n x n* existirão *n* vectores próprios ou não existirá nenhum.

Todos os vectores próprios de uma matriz são perpendiculares, ou seja, definem ângulos rectos entre si, independentemente do número de dimensões do conjunto.

A expressão $|M - \lambda I| = 0$, designada equação característica de M, permite o cálculo dos valores próprios da matriz. Refira-se que I é a matriz identidade de dimensão *n*. Os vectores próprios de M associados ao valor próprio λ são as soluções do sistema $(M - \lambda I)\vec{VP} = 0$.

4.2 DESCRIÇÃO DO MÉTODO

A aplicação do método de Análise de Componentes Principais sobre dados estatísticos permite evidenciar padrões na sua informação. Estes padrões podem ser usados na compressão dos dados, reduzindo para tal o número de dimensões que o compõem, sem perda significativa de informação. Esta técnica é vulgarmente utilizada na compressão de imagem.

A seguir são descritos os vários passos do método.

SUBTRACÇÃO DA MÉDIA

Para obter os resultados desejados é necessário subtrair a média de cada dimensão a cada ponto que a compõe:

$$M_{Sub} = \begin{pmatrix} x_1 - \overline{x} & y_1 - \overline{y} & z_1 - \overline{z} \\ x_2 - \overline{x} & y_2 - \overline{y} & z_2 - \overline{z} \\ \dots & \dots & \dots \\ x_n - \overline{x} & y_n - \overline{y} & z_n - \overline{z} \end{pmatrix}$$

Desta operação resulta um conjunto de dados com média 0.

CÁLCULO DA MATRIZ DE COVARIÂNCIA

Com os dados resultantes da subtracção, calcula-se a matriz de covariância.

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

CÁLCULO DOS VECTORES E VALORES PRÓPRIOS

Devido ao facto da matriz de covariância ser quadrada, é possível calcular os vectores próprios e valores próprios associados. Como referido anteriormente, os vectores próprios permitem caracterizar o comportamento dos dados.

ESCOLHA DAS COMPONENTES PRINCIPAIS

Após a obtenção dos vectores e valores próprios, é chegada a altura de comprimir/reduzir a dimensão do conjunto de dados.

Quanto maior for o valor próprio associado ao vector próprio da matriz de covariância maior será a importância do vector próprio na caracterização do conjunto de dados. Se ignorarmos os vectores próprios menos significativos (valores próprios menores), consegue-se reduzir o número de dimensões sem grande perda de informação.

O passo seguinte é colocar os vectores próprios numa matriz:

$$M_{\vec{VP}} = \begin{pmatrix} \vec{VP}_1 & \vec{VP}_2 & \dots & \vec{VP}_n \end{pmatrix}$$

CÁLCULO DO CONJUNTO DE DADOS FINAL

Este é o último passo do método *PCA*. Uma vez escolhidas as componentes que pretendemos manter e construída a sua matriz, podemos calcular a matriz com os dados originais, sem a informação relativa aos vectores próprios menos significativos, da seguinte forma:

$$M_{PCA} = M^{T}_{\vec{VP}} \times M_{Sub}$$

5 APRESENTAÇÃO DE RESULTADOS

Neste capítulo apresentam-se os resultados obtidos nos ensaios realizados para cada uma das aplicações, dentro de cenários semelhantes.

Esta premissa facilitou a análise gráfica bidimensional que foi usada para a identificação dos parâmetros retornados pelo *TSTAT* considerados mais relevantes para a análise de componentes principais.

Por último, é descrito o processo usado para a identificação de características únicas no tráfego de uma aplicação.

5.1 DESCRIÇÃO DOS CENÁRIOS DE TESTE

Entre Março e Julho de 2007 foi realizado um conjunto de ensaios de geração de tráfego através de aplicações *P2P* que teve por objectivo a recolha de dados estatísticos para a identificação de características únicas de cada aplicação.

De forma a manter a integridade das capturas, foram mantidas configurações semelhantes entre as diversas aplicações, bem como ficheiros partilhados e definições de rede. A ferramenta *WinDump* permitiu o registo integral dos pacotes recebidos/enviados por cada aplicação.

Para melhor comparação entre as aplicações do mesmo tipo foram realizados testes de características semelhantes: configurações, duração, pesquisas e ficheiros transferidos. De notar que no arranque de cada uma das aplicações não existiam ficheiros partilhados.

Os testes realizados encontram-se descritos nas sub-secções seguintes.

5.1.1 SHAREAZA

A sessão *Shareaza* usada para captura de tráfego durou aproximadamente uma hora e consistiu no seguinte:

1. Início da captura de tráfego gerado e recebido pela máquina (*windump -i 2 -C 700 -w "shareazaCapture.dmp"*);

2. Arranque da aplicação *Shareaza*. Verificar que a máquina se ligou a pelo menos outras 4 máquinas existentes na rede *Gnutella*;
3. Aguardar cerca de 2 minutos para estabilização da ligação da máquina à rede *Gnutella* (fluxo de mensagens de inicialização da aplicação);
4. Pesquisa por "***Vista Transformation Pack***", durante 1 minuto;
5. Pedido de transferência do ficheiro "***Vista Transformation Pack 6.0.exe***" (30.26MB);
6. Passados 2 minutos realizar pesquisa por "***Skype***", durante de 1 minuto;
7. Pedido de transferência do ficheiro "***SkypeSetup.exe***" (12.25MB);
8. Desligar a aplicação, após 10 minutos de actividade;
9. Abertura da aplicação. Aguardar 2 minutos para estabilização da ligação da máquina à rede *Gnutella* (verificar efeito do re-arranque da aplicação);
10. Pesquisa por "***AVG***" e ficheiros com tamanho mínimo de 10MB, durante 1 minuto;
11. Pedido de transferência do ficheiro "***AVG 7.5.exe***" (16.70MB);
12. Passados 5 minutos realizar pesquisa por "***Ubuntu***", durante 1 minuto;
13. Pedido de transferência do ficheiro "***ubuntu-6.10-desktop-i386.iso***" (698.37MB);
14. Encerramento da aplicação, após 1 hora de actividade;
15. Fim da captura de tráfego.

O ficheiro resultante da captura foi processado através da ferramenta *TSTAT*, para recolha dos dados estatísticos. Esta informação foi posteriormente usada na identificação de padrões de tráfego gerado pela aplicação *Shareaza*.

5.1.2 EMULE

A sessão *eMule* usada para captura de tráfego durou aproximadamente uma hora e consistiu no seguinte:

1. Início da captura de tráfego gerado e recebido pela máquina (*windump -i 2 -C 700 -w "emuleCapture.dmp"*);
2. Arranque da aplicação *eMule* e ligação ao servidor "***DonkeyServer No2***";

3. Aguardar cerca de 2 minutos para estabilização da ligação da máquina à rede *eDonkey2000* (fluxo de mensagens de inicialização da aplicação);
4. Pesquisa por "***Vista Transformation Pack***", durante 1 minuto;
5. Pedido de transferência do ficheiro "***Vista Transformation Pack.exe***" (30.27MB);
6. Passados 2 minutos realizar pesquisa por "***Skype***", durante de 1 minuto;
7. Pedido de transferência do ficheiro "***Skype V. 2.5.0.130.exe***" (9.83MB);
8. Mudança para o servidor "***DonkeyServer No1***", após 10 minutos de actividade. Aguardar 2 minutos para estabilização da ligação da máquina à rede *eDonkey2000* (verificar efeito da troca de servidor);
9. Pesquisa por "***AVG***" e ficheiros com tamanho mínimo de 10MB, durante 1 minuto;
10. Pedido de transferência do ficheiro "***AVG.Antivirus.plus.Firewall.Edition.rar***" (16.75MB);
11. Passados 5 minutos realizar pesquisa por "***Ubuntu***", durante 1 minuto;
12. Pedido de transferência do ficheiro "***Ubuntu-6.10-Desktop-i386.iso***" (698.37MB);
13. Encerramento da aplicação, após 1 hora de actividade;
14. Fim da captura de tráfego.

O ficheiro resultante da captura foi processado através da ferramenta *TSTAT*, para recolha dos dados estatísticos. Esta informação foi posteriormente usada na identificação de padrões de tráfego gerado pela aplicação *eMule*.

5.1.3 BITTORRENT

A sessão *bitTorrent* usada para captura de tráfego durou aproximadamente uma hora e consistiu no seguinte:

1. Início da captura de tráfego gerado e recebido pela máquina (*windump -i 2 -C 700 -w "bittorrentCapture.dmp"*);
2. Ligação ao *website* ***mininova.org***:
    a. Pesquisa por "***Vista Transformation Pack***";

b. Pedido de transferência do *torrent* "***Vista Transformation Pack 6.0.exe.torrent***" (9.8KB);

3. Arranque da aplicação *bitTorrent*;
4. Aguardar cerca de 2 minutos para estabilização da ligação da máquina à rede *bitTorrent* (fluxo de mensagens de inicialização da aplicação);
5. Iniciar transferência do ficheiro "***Vista Transformation Pack 6.0.exe***" (30.27MB), usando o *torrent*;
6. Passados 2 minutos, retomar a ligação ao *website mininova.org*:
    a. Pesquisa por "***Skype***";
    b. Pedido de transferência do *torrent* "***Skype V. 3.1.0.152 Final.exe.torrent***" (8.4KB);
7. Iniciar a transferência do ficheiro "***Skype V. 3.1.0.152 Final.exe***" (19.97MB), usando o *torrent*;
16. Desligar a aplicação, após 10 minutos de actividade;
8. Abertura da aplicação. Aguardar 2 minutos para estabilização da ligação da máquina à rede *bitTorrent* (verificar efeito do re-arranque da aplicação);
9. Retomar ligação ao *website mininova.org*:
    a. Pesquisa por "***AVG***";
    b. Pedido de transferência do *torrent* "***AVG Professional Internet Security Suite.rar.torrent***" (10.6KB);
10. Iniciar a transferência do ficheiro "***AVG Professional Internet Security Suite.rar***" (39.23MB), usando o *torrent*;
11. Passados 5 minutos, retomar a ligação ao *website mininova.org*:
    a. Pesquisa por "***Ubuntu***";
    b. Pedido de transferência do *torrent* "***Ubuntu-6.10-Desktop-i386.iso.torrent***" (27.5KB);
12. Iniciar a transferência do ficheiro "***Ubuntu-6.10-Desktop-i386.iso***" (698.36MB), usando o *torrent*;
15. Encerramento da aplicação, após 1 hora de actividade;
16. Fim da captura de tráfego.

O ficheiro resultante da captura foi processado através da ferramenta *TSTAT*, para recolha dos dados estatísticos. Esta informação foi posteriormente usada na identificação de padrões de tráfego gerado pela aplicação *bitTorrent*.

5.1.4 HTTP

A sessão *HTTP* (servidor *webbased*) usada para captura de tráfego durou aproximadamente uma hora e consistiu no seguinte:

1. Início da captura de tráfego gerado e recebido pela máquina (*windump -i 2 -C 700 -w "httpCapture.dmp"*);
2. Ligação ao *website* ***download.com***;
3. Pesquisa por "***Vista Transformation Pack***";
4. Pedido de transferência do ficheiro "***Vista_Transformation_Pack_6.0.exe***" (30.27MB);
5. Passados cerca de 2 minutos, pesquisar por "***Skype***";
6. Pedido de transferência do ficheiro "***SkypeSetup.exe***" (9.83MB);
7. Dois minutos após o início da transferência, pesquisar por "***AVG***";
8. Pedido de transferência do ficheiro "***avg75_488a1138.rar***" (27.2MB);
9. Passados 5 minutos, pesquisar por "***Ubuntu***";
10. Pedido de transferência do ficheiro "***Ubuntu-6.10-Desktop-i386.iso***" (697MB);
11. Fim da captura de tráfego, após 1 hora de actividade.

O ficheiro resultante da captura foi processado através da ferramenta *TSTAT*, para recolha dos dados estatísticos. Esta informação foi posteriormente usada na identificação de padrões de tráfego gerado pela transferência de ficheiros a partir do *website download.com*.

5.1.5 BROWSING

A sessão de Navegação usada para captura de tráfego durou aproximadamente uma hora e consistiu no seguinte:

1. Início da captura de tráfego gerado e recebido pela máquina (*windump -i 2 -C 700 -w "browsingCapture.dmp"*);
2. Ligação ao *website* ***gmail.com***;

3. Entrada na conta de *e-mail* e leitura de algumas mensagens, durante cerca de 5 minutos;
4. Ligação ao *website* ***google.com***;
5. Pesquisa por "**publico.pt**";
6. Ligação ao *website* ***publico.clix.pt***;
7. Navegação pelas notícias do dia, durante cerca de 10 minutos;
8. Ligação ao *website* ***jornaldenegocios.pt***;
9. Navegação pelas notícias económicas do dia, durante cerca de 10 minutos;
10. Ligação ao *website* ***exameinformatica.clix.pt***;
11. Navegação pelos artigos existentes, durante cerca de 10 minutos;
12. Ligação ao *website* ***pcguia.xl.pt***;
13. Navegação pelos artigos existentes, durante cerca de 10 minutos;
14. Ligação ao *website* ***gizmodo.com***;
15. Navegação pelos artigos existentes, durante cerca de 10 minutos;
16. Fim da captura de tráfego, após 1 hora de actividade.

O ficheiro resultante da captura foi processado através da ferramenta *TSTAT*, para recolha dos dados estatísticos. Esta informação foi posteriormente usada na identificação de padrões de tráfego gerado pela navegação em *websites* Internet.

5.1.6 STREAMING

A sessão de *Streaming* usada para captura de tráfego durou aproximadamente uma hora e consistiu na audição da emissão *online* da Rádio Renascença, disponível no *website* ***rr.pt*** a *32Kbps*.

O ficheiro resultante da captura foi processado através da ferramenta *TSTAT*, para recolha dos dados estatísticos. Esta informação foi posteriormente usada na identificação de padrões de tráfego gerado pelo *Streaming* Internet.

5.1.7 SKYPE

A sessão *Skype* usada para captura de tráfego durou aproximadamente uma hora e consistiu na conversa telefónica *VOIP* entre dois utilizadores, sendo que um deles iniciou a aplicação na máquina monitorada.

O ficheiro resultante da captura foi processado através da ferramenta *TSTAT*, para recolha dos dados estatísticos. Esta informação foi posteriormente usada na identificação de padrões de tráfego gerado pela aplicação *Skype*.

5.1.8 YOUTUBE

A sessão *youTube* usada para captura de tráfego durou aproximadamente uma hora e consistiu no seguinte:

1. Início da captura de tráfego gerado e recebido pela máquina (*windump -i 2 -C 700 -w "bittorrentCapture.dmp"*);
2. Ligação ao *website* ***youtube.com***;
3. Pesquisa por "**gato fedorento**";
4. Ordenação por "*Rating*" dos vídeos que respeitam a pesquisa;
5. Abertura e visionamento do vídeo "**Gato Fedorento – Tesourinhos Deprimentes: Mistura Fina**";
6. Abertura e visionamento do vídeo "**Gato Fedorento – Presidente da Camara de VN da Rabona**";
7. Abertura e visionamento do vídeo "**Gato Fedorento – A minha vida dava um filme indiano**";
8. Abertura e visionamento do vídeo "**Gato Fedorento – O Homem é Maior de sua Aldeia**";
9. Abertura e visionamento do vídeo "**Gato Fedorento – Corrupção em Portugal**";
10. Abertura e visionamento do vídeo "**Gato Fedorento – Alberto Joao Jardim**";
11. Abertura e visionamento do vídeo "**Gato Fedorento – As Vindimas**";
12. Abertura e visionamento do vídeo "**Gato Fedorento – Filosofo Matarruano**";
13. Abertura e visionamento do vídeo "**Gato Fedorento – Escuteiros (Perfeito Anormal)**";
14. Abertura e visionamento do vídeo "**Gato Fedorento – Agricultor**";
15. Abertura e visionamento do vídeo "**Gato Fedorento – As grandes questões do nosso tempo**";
16. Fim da captura de tráfego, após 1 hora de actividade.

O ficheiro resultante da captura foi processado através da ferramenta *TSTAT*, para recolha dos dados estatísticos. Esta informação foi posteriormente usada na identificação de padrões de tráfego gerado pela navegação no *website youTube*.

5.2 RESULTADOS GLOBAIS

Cada um dos ensaios descritos no ponto anterior originou um ficheiro com os dados da captura realizada. As tabelas seguintes apresentam o resultado do processamento *TSTAT* para cada um desses ficheiros, bem como o tempo médio de cada sessão.

	CLIENT								
	Packets	RST	ACK	Pure ACK	KBytes	Data packets	Kbytes (w/ retrans)	rexmit packets	rexmit KBytes
Shareaza	58893	65	56077	53225	145,072	2761	157,292	1847	13,965
eMule	116631	16	115376	21930	107780,784	92451	108433,256	1394	652,833
bitTorrent	205082	174	203373	85392	124903,684	117242	126209,951	2675	1306,845
Http	79923	281	79192	76818	1120,075	1723	1126,331	42	6,29
Browsing	11358	317	10450	7364	1225,208	2351	1296,95	143	71,826
Streaming	10062	3	9833	9818	4,113	8	4,113	141	0,141
Skype	318	3	226	59	4,491	150	4,499	46	0,052
youTube	40670	170	40383	38206	1369,82	1912	1401,508	61	31,698
	SYN count	FIN count	SACK sent	rtx RTO	rtx FR	reordering	unknown	unnece rtx RTO	
Shareaza	2815	27	5463	1824	14	0	21	0	
eMule	1251	1002	1918	1146	21	1694	1563	128	
bitTorrent	1679	646	4925	1503	117	3735	2159	861	
Http	730	371	383	42	0	0	0	0	
Browsing	908	418	249	133	1	0	9	0	
Streaming	229	4	0	141	0	0	0	0	
Skype	92	14	7	43	0	0	3	0	
youTube	287	95	1153	33	0	0	28	0	

Tabela 2: Totais de *Upload*

	SERVER								
	Packets	RST	ACK	Pure ACK	KBytes	Data packets	Kbytes (w/ retrans)	rexmit packets	rexmit KBytes
Shareaza	93273	17	93269	2068	115106,694	91029	115244,894	257	138,212
eMule	91451	21	91446	55717	34366,424	33826	35017,843	954	651,495
bitTorrent	216601	270	216574	80998	129338,573	133337	131278,927	3164	1940,692
Http	151435	8	151430	1315	205121,848	149113	205184,553	65	62,717
Browsing	14133	22	14126	1443	11368,405	11538	11377,214	39	8,838
Streaming	19626	0	19626	5	15459,787	19610	15459,787	0	0
Skype	343	0	343	138	37,696	184	37,696	1	0,001
youTube	72746	7	72740	445	96745,717	71934	97004,513	194	258,8
	SYN count	FIN count	SACK sent	rtx RTO	rtx FR	reordering	unknown	unnece rtx RTO	
Shareaza	115	54	4	86	4	858	440	99	
eMule	959	935	4087	660	15	187	978	224	
bitTorrent	1138	931	6717	2416	213	1478	2616	514	
Http	674	573	0	55	32	176	34	6	
Browsing	787	577	1	68	32	136	44	1	
Streaming	7	4	0	0	0	0	2	0	
Skype	11	11	0	1	0	9	1	0	
youTube	272	163	0	30	30	776	123	4	

Tabela 3: Totais de *Download*

Mean Duration (sec)	Shareaza	eMule	bitTorrent	Http	Browsing	Streaming	Skype	youTube
	31,83	79,25	108,84	24,65	28,25	50,54	70,89	67,11

Tabela 4: Duração média de cada sessão, por aplicação

De entre os parâmetros apresentados, o tamanho dos pacotes transmitido em *KBytes* oferece-nos uma medida do tráfego gerado por cada aplicação: como esperado tanto no *upload* como no *download* existe uma predominância das aplicações de partilha de ficheiros, existindo contudo algumas excepções que merecem referência.

O *upload* gerado através da aplicação *Shareaza* é significativamente inferior ao gerado através das restantes aplicações *P2P* de partilha de ficheiros, o que evidencia uma das características apontadas ao algoritmo do protocolo *Gnutella*: só existe partilha de ficheiros completos.

Já no *download*, o *eMule* destaca-se por ser a aplicação de partilha de ficheiros com menor rendimento, tendo mesmo um valor inferior ao *upload* no ensaio realizado.

Neste aspecto as aplicações mais eficientes foram o *Shareaza* e os servidores *webbased* (*HTTP*).

Para a restrição/filtragem do tráfego capturado foi decidido que apenas seria observado o tráfego *TCP* gerado e recebido pela máquina. Este facto estará na origem dos resultados obtidos para o ensaio realizado com a aplicação *Skype*, onde o número de fluxos *TCP* observado é pouco significativo. Isto indicia que grande parte da comunicação vocal é feita via *UDP*.

5.3 ANÁLISE GRÁFICA

O **MATLAB** é uma ferramenta matemática poderosa que dispõe de um extenso conjunto de funções e uma linguagem de programação de alto nível.

Foi gerado um conjunto de *m-files* (ficheiros que permitem a submissão de uma sequência de comandos MATLAB de uma só vez) para o processamento da informação proveniente das capturas. Uma das funções desenvolvidas constrói automaticamente gráficos bidimensionais entre os parâmetros disponibilizados pela aplicação *TSTAT*.

A partir dos gráficos foi possível identificar relações entre parâmetros, bem como o relevo de cada um deles na caracterização do protocolo. Do conjunto final de parâmetros usado na caracterização de tráfego foram escolhidos para apresentação na dissertação aqueles que graficamente apresentaram maiores diferenças entre os vários protocolos. Compõem este conjunto os seguintes parâmetros: Duração do Fluxo, Total de Pacotes, Total de mensagens do tipo ACK, Total de *Bytes* enviados no *payload* das mensagens,

Total de mensagens do tipo SYN, Tempo de *Round Trip* válido e Tempo médio de *Round Trip*.

Seguidamente apresentam-se alguns dos gráficos possíveis para cada par de parâmetros do conjunto final.

5.3.1 DURAÇÃO DO FLUXO

As figuras seguintes apresentam a variação dos parâmetros escolhidos em função da duração de cada fluxo.

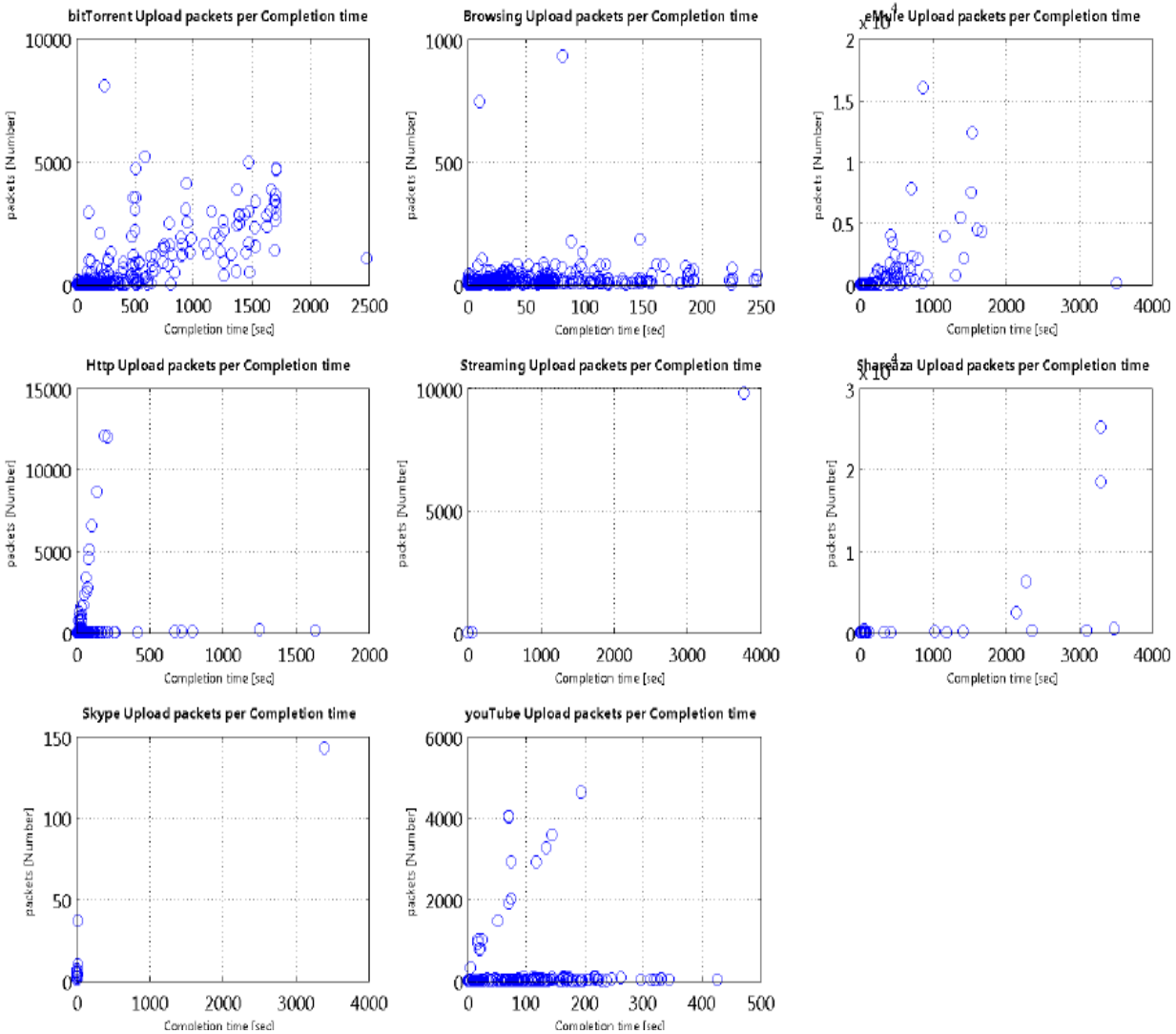

figura 1: *Packets Upload per Completion Time*

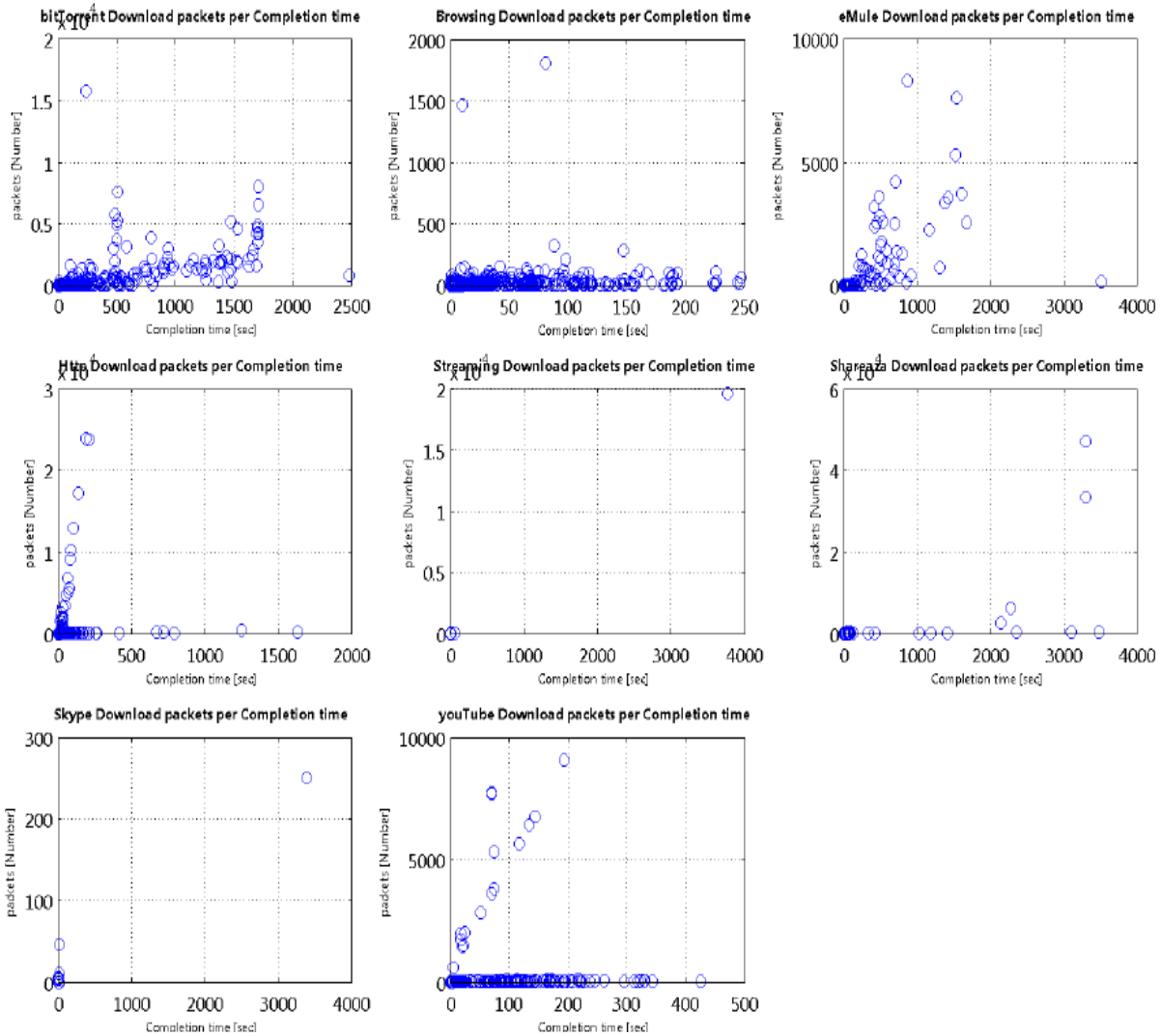

figura 2: *Packets Download per Completion Time*

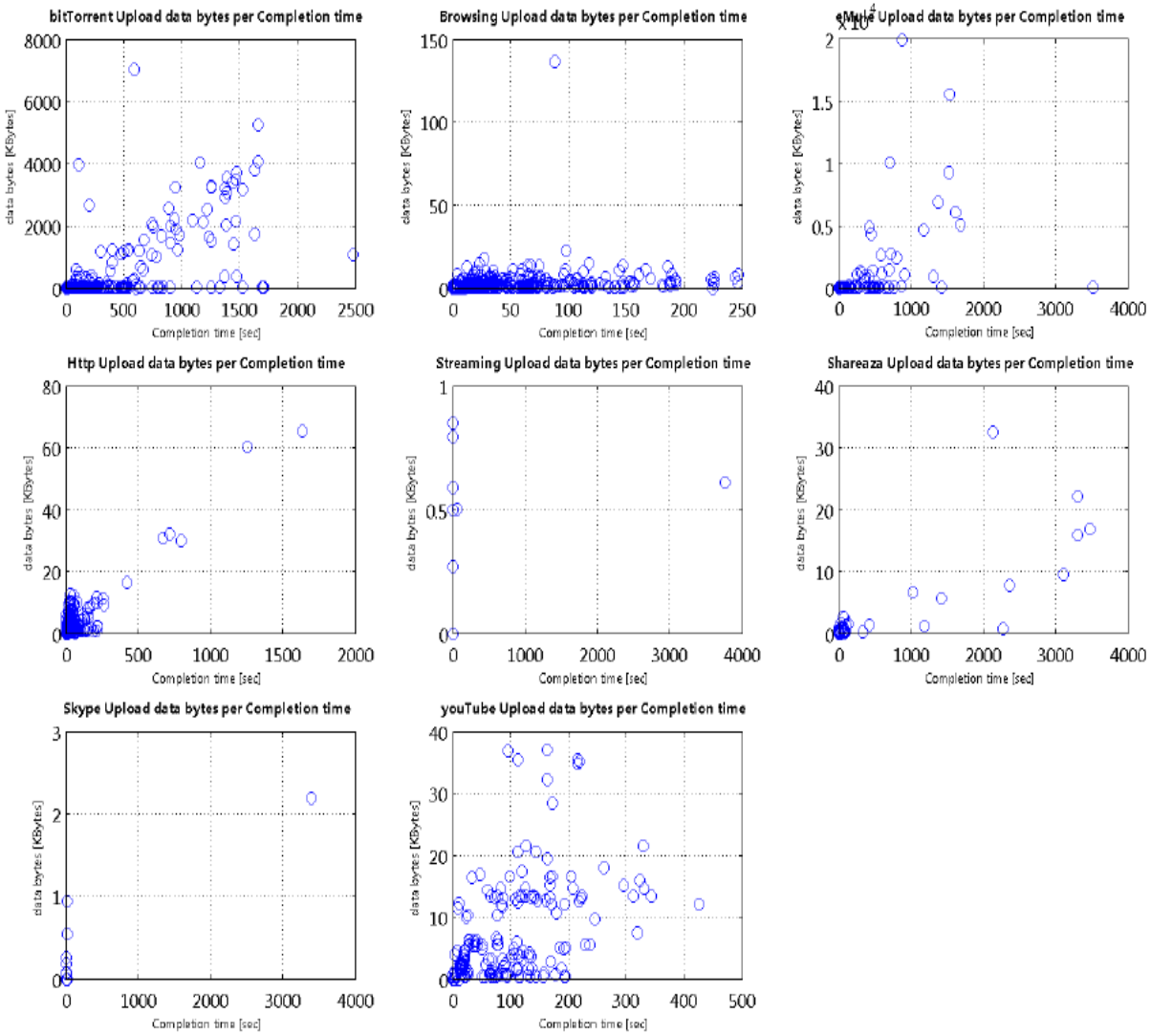

figura 3: *Data Bytes Upload per Completion Time*

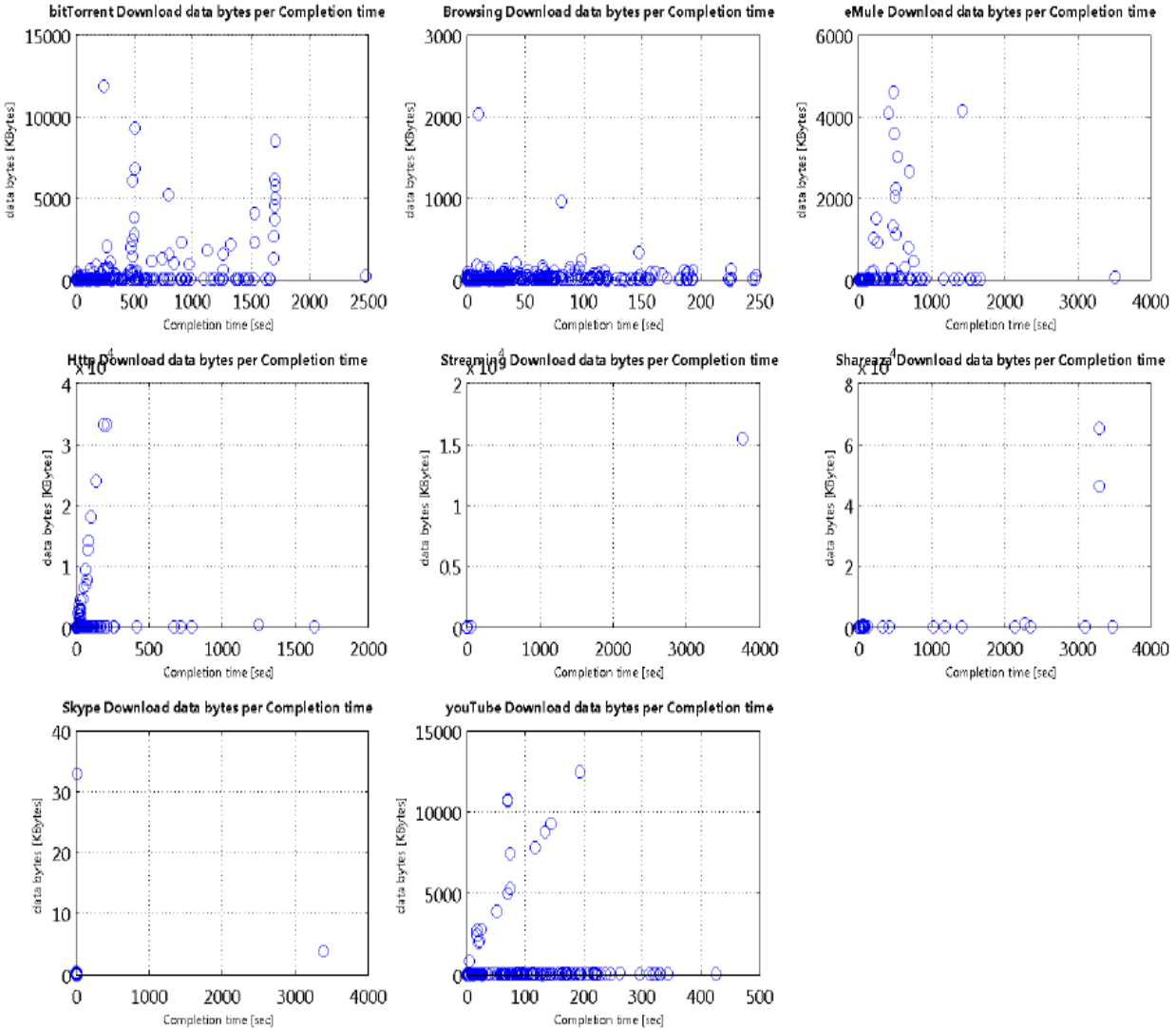

figura 4: *Data Bytes Download per Completion Time*

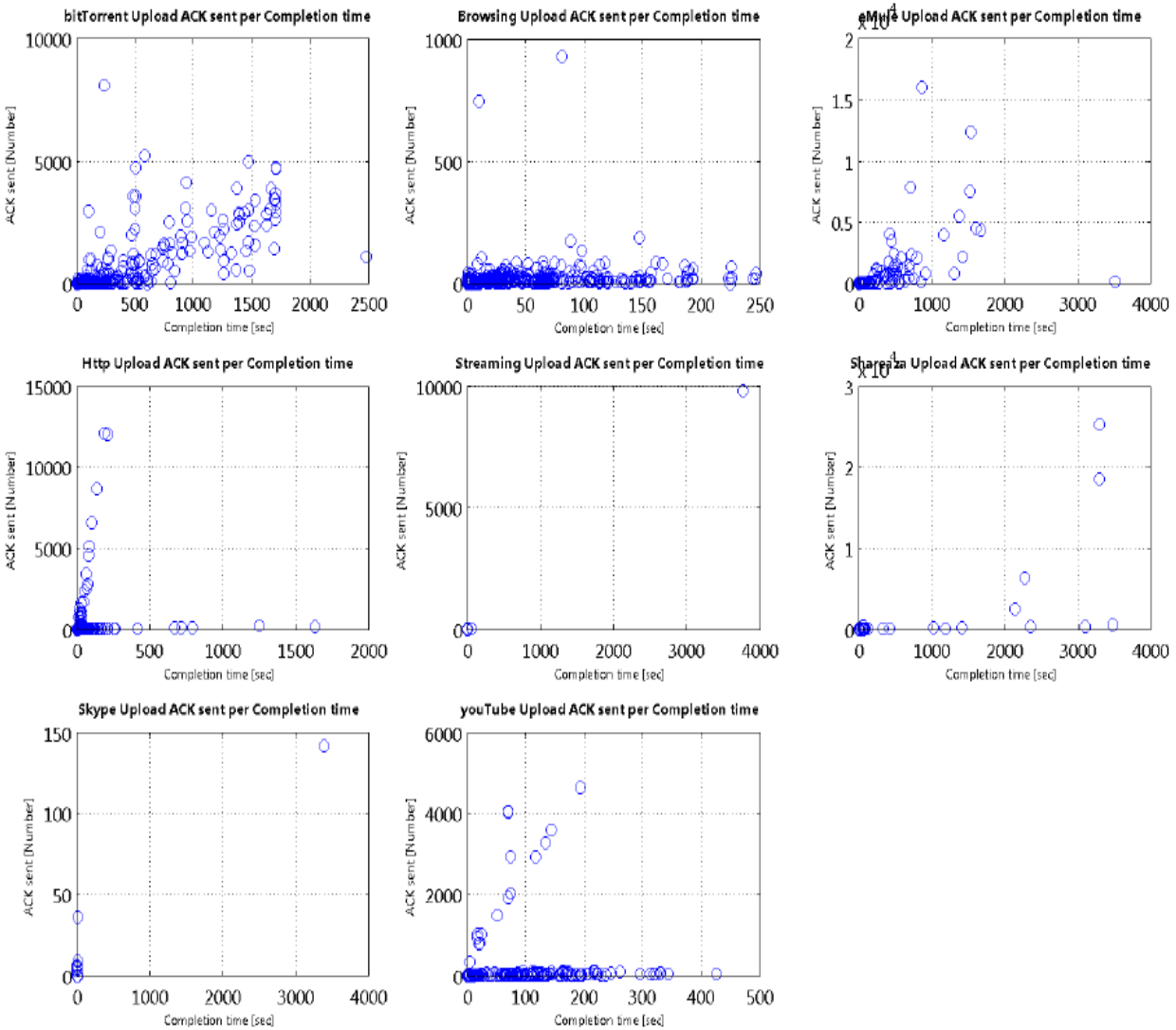

figura 5: *ACK Messages Upload per Completion Time*

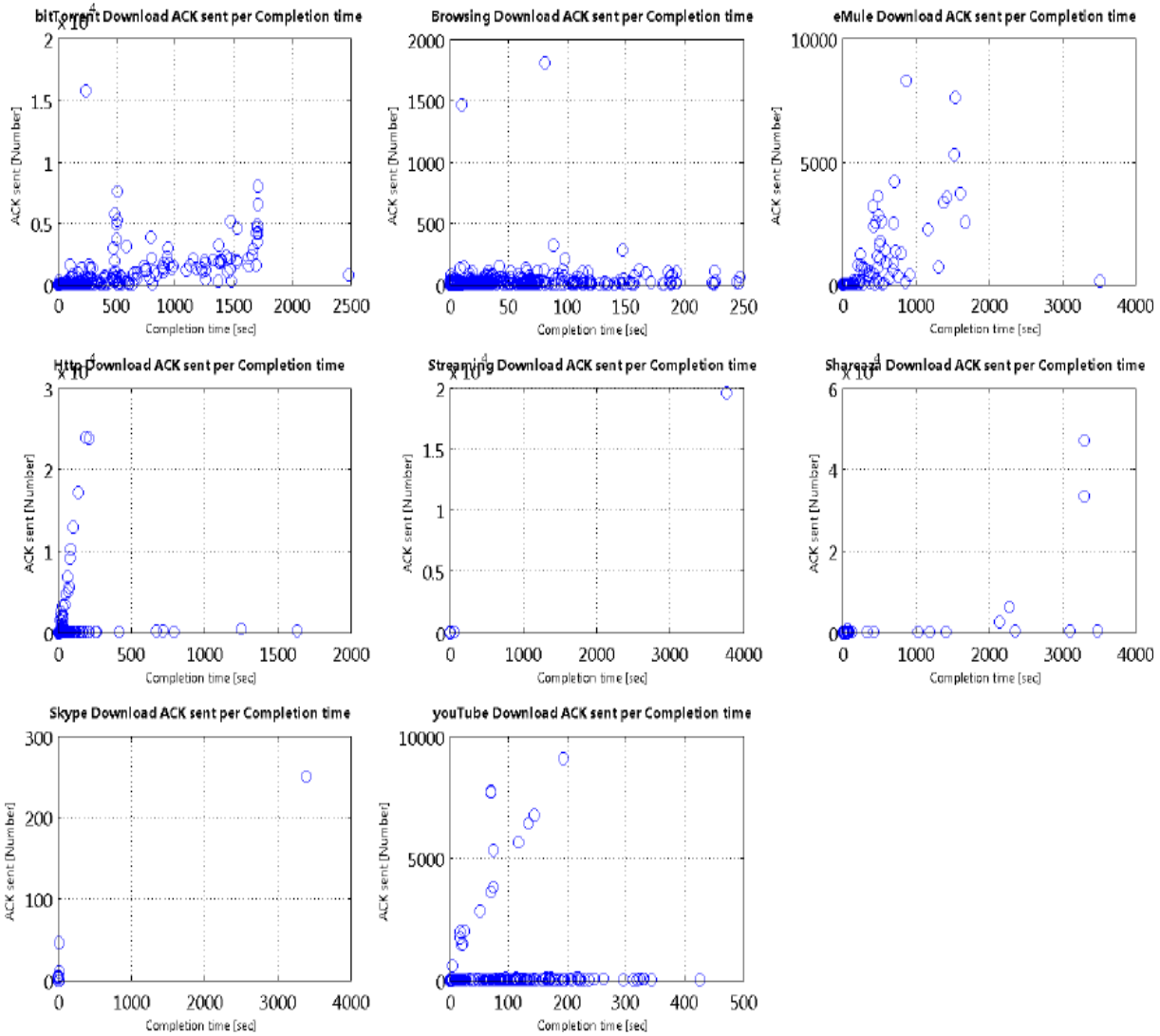

figura 6: *ACK Messages Download per Completion Time*

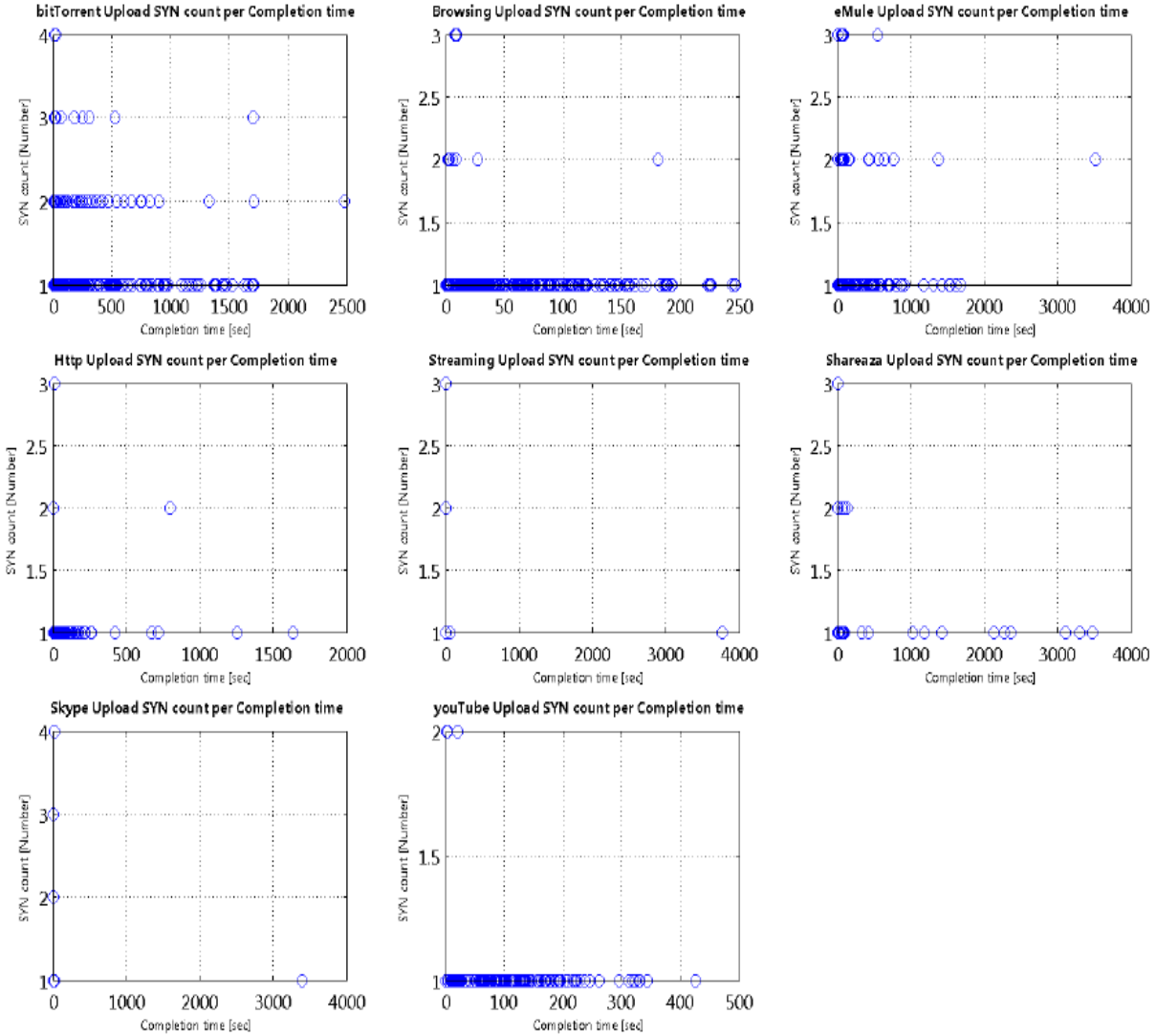

figura 7: *SYN Messages Upload per Completion Time*

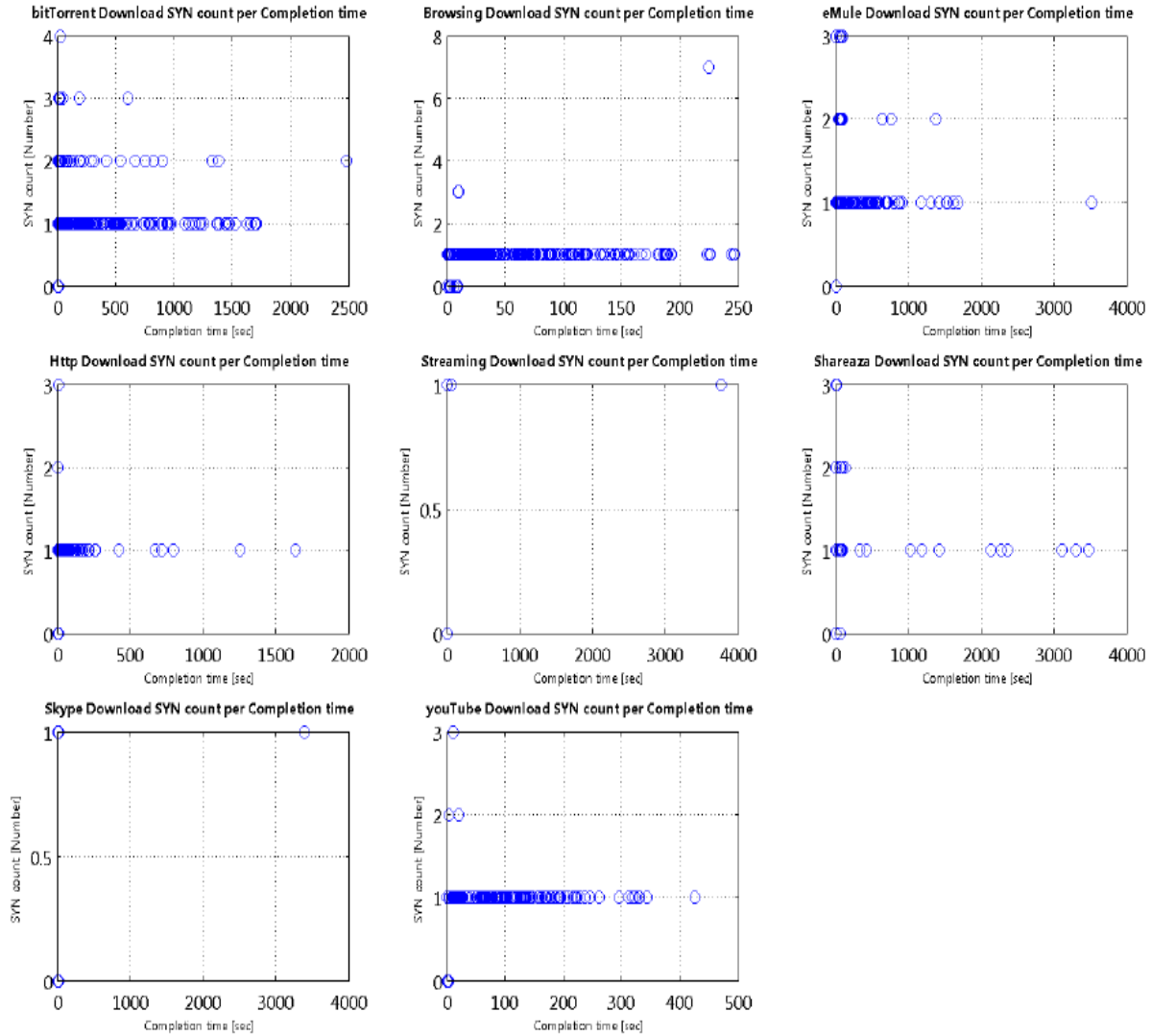

figura 8: *SYN Messages Download per Completion Time*

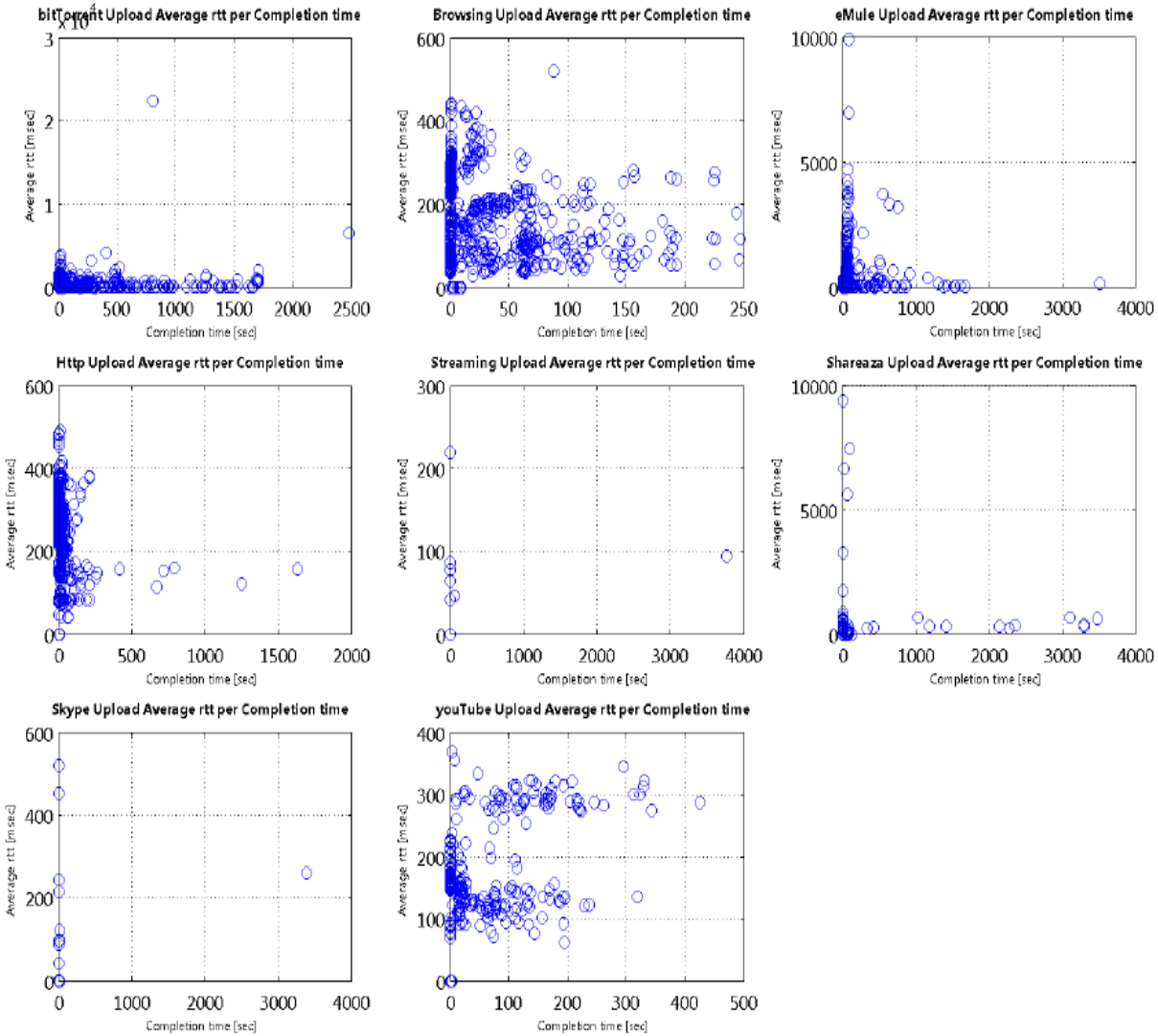

figura 9: *Average Round Trip Time Upload per Completion Time*

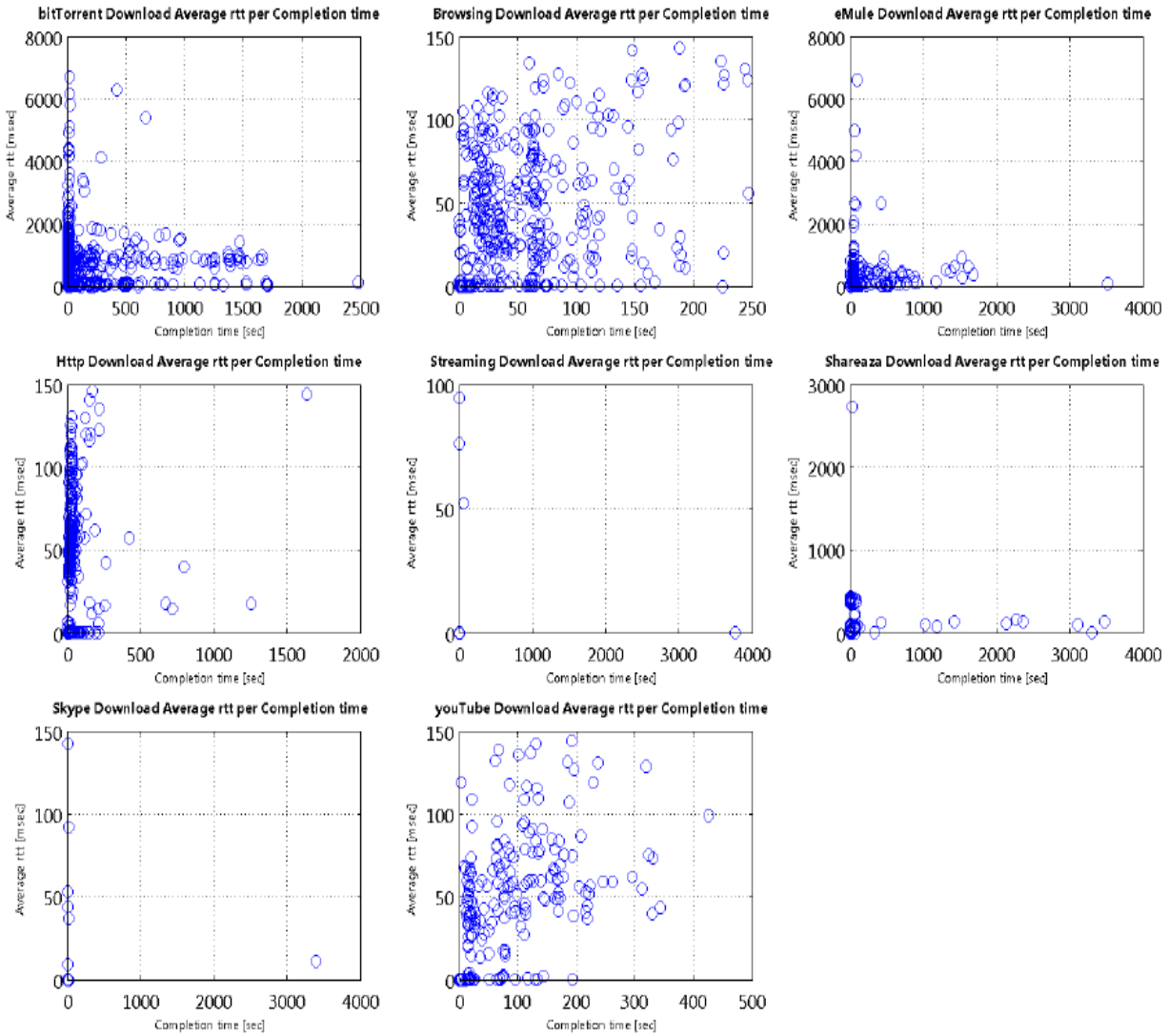

figura 10: *Average Round Trip Time Download per Completion Time*

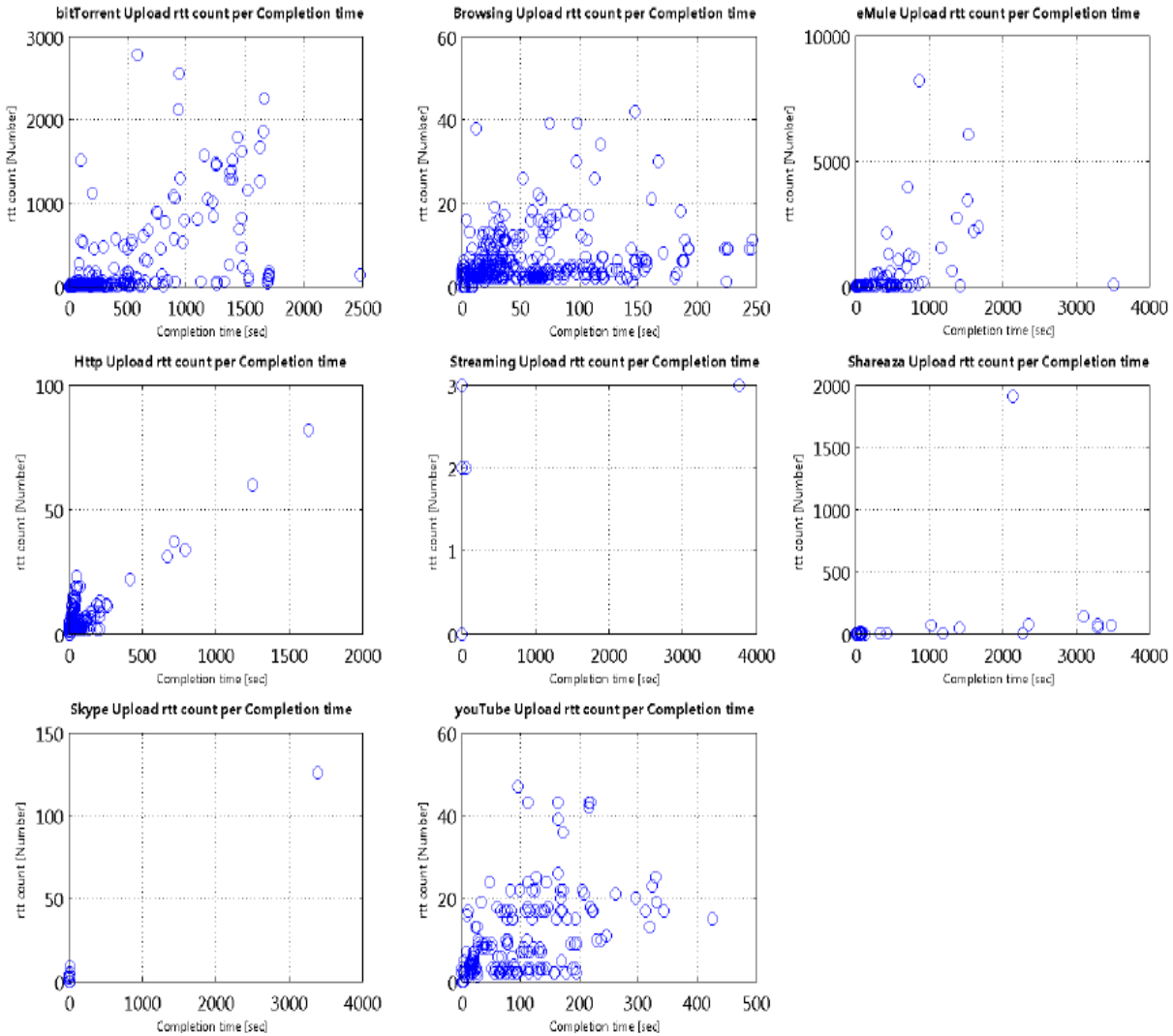

figura 11: *Valid Round Trip Time Upload per Completion Time*

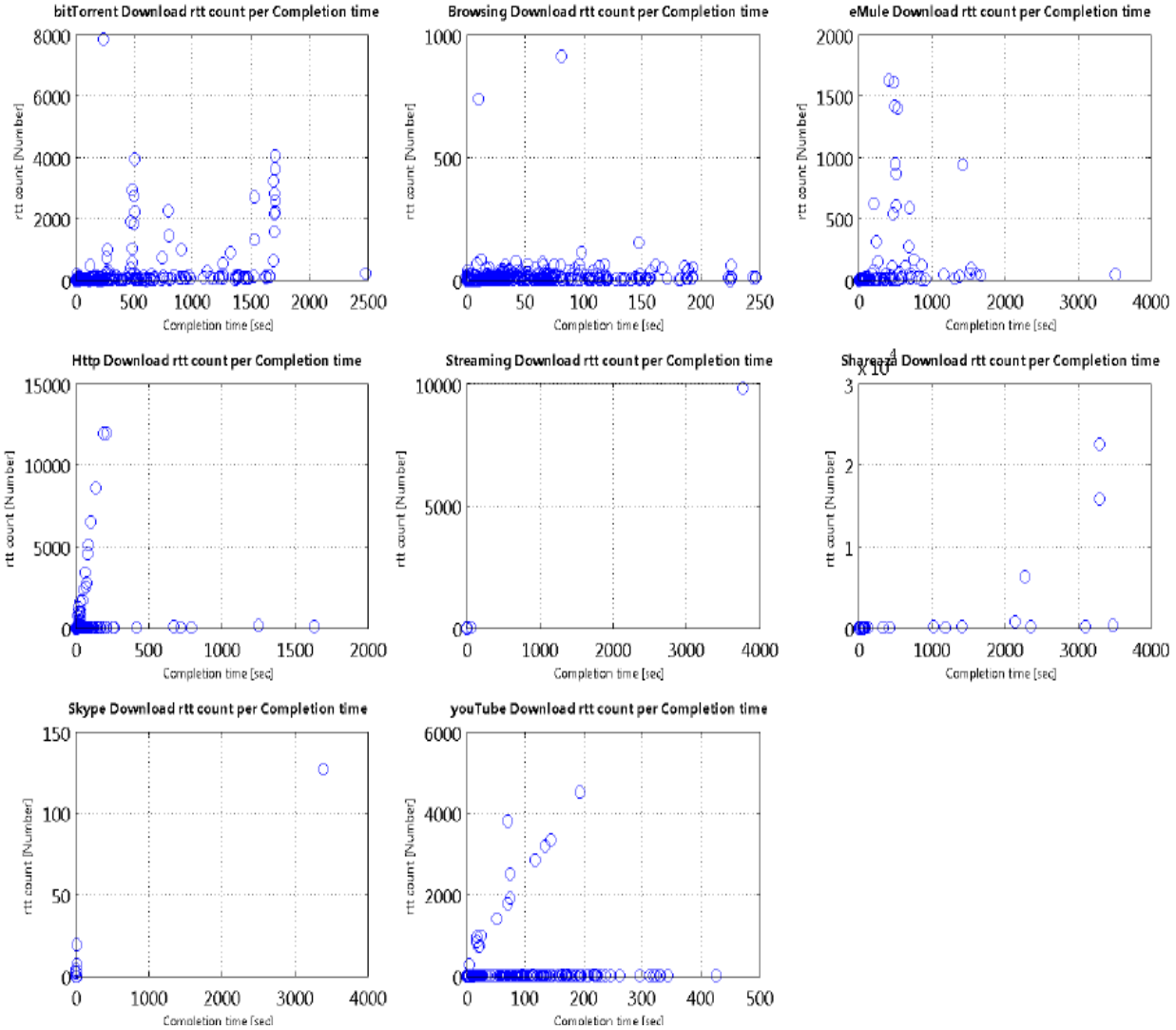

figura 12: *Valid Round Trip Time Download per Completion Time*

Os gráficos *bitTorrent* apresentam um crescimento linear no que diz respeito ao número de pacotes/*KBytes* em função da duração do fluxo, tratando-se de um denominador comum às restantes aplicações de partilha de ficheiros. Contudo, verifica-se também que o fluxo com maior número de pacotes transferido/recebido é de curta duração. A fragmentação dos ficheiros estará na origem destas excepções à linearidade: a velocidade de transferência é tanto maior quanto maior for a largura de banda do par. Neste caso, estaremos a falar de um par com uma largura de banda e recursos de elevado desempenho.

Já no que respeita ao protocolo *Shareaza*, as transferências assentam num modelo cliente/servidor em que apenas se transferem ficheiros completos, pelo que a diferença de recursos entre os pares não é tão significativa.

Os gráficos obtidos para os *downloads* através de servidores *webbased* (*HTTP*), *youTube* e *Streaming* apresentam uma linearidade vincada, o que se justifica pela arquitectura cliente/servidor em que assentam (os recursos mantêm-se ao longo da actividade).

O número de pacotes/*KBytes* transferidos no *Browsing*, mais do que em qualquer outra aplicação analisada, estão dependentes não só do servidor que dispõe dos conteúdos pretendidos mas também do tamanho dos mesmos. Em geral, tratam-se de conteúdos de tamanho reduzido (centenas de *KBytes*), quando comparados com os transferidos pelas restantes aplicações (*MBytes*). Por este motivo, é natural que seja a aplicação que registe o maior número de fluxos, para 1 hora de actividade (duração das sessões de medição efectuadas).

5.3.2 TEMPO DE *ROUND TRIP* VÁLIDO

As figuras seguintes apresentam a variação dos parâmetros escolhidos em função do parâmetro correspondente às contagens válidas de tempo entre o envio de uma mensagem e recepção da mensagem do tipo ACK correspondente.

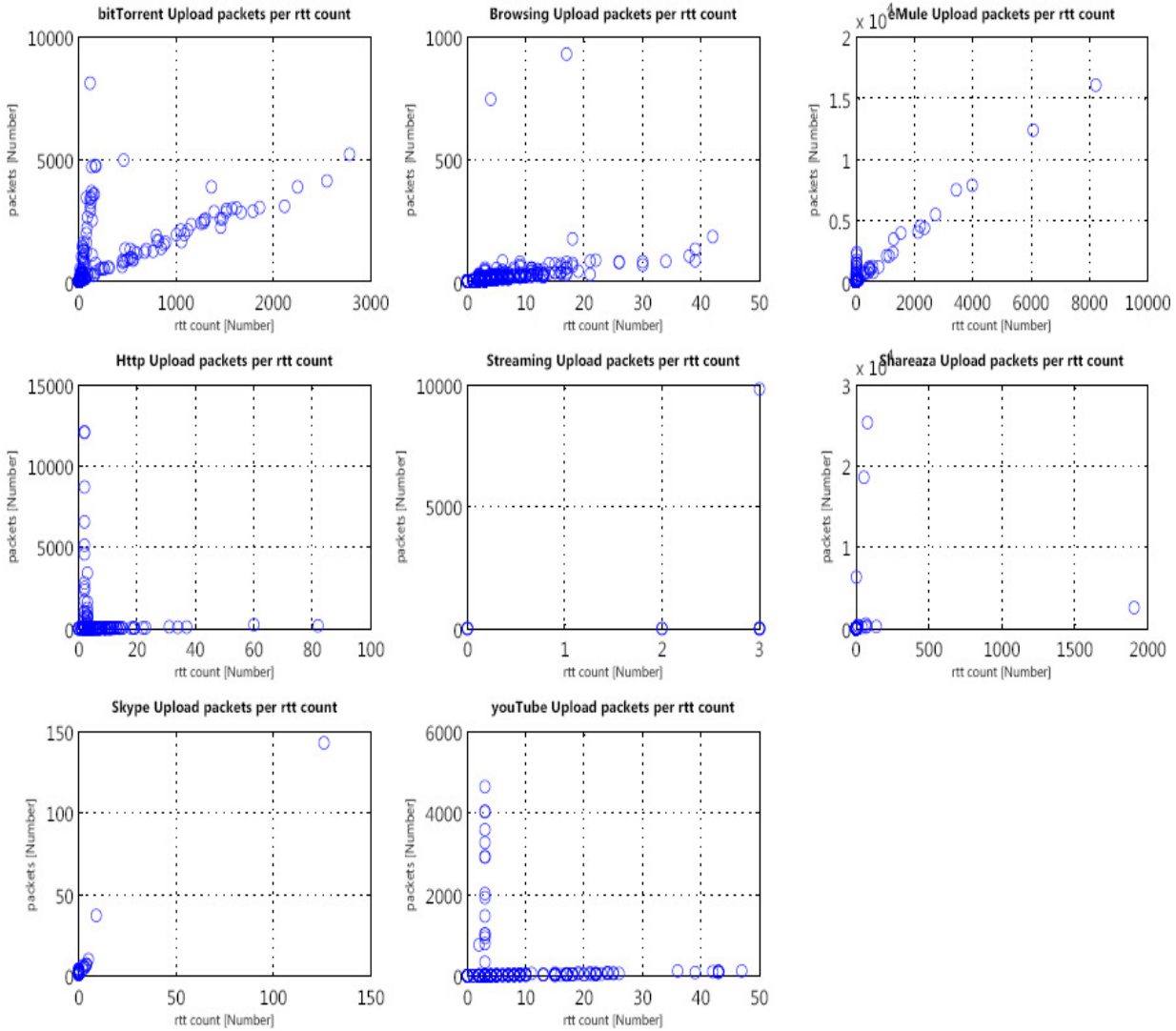

figura 13: *Packets Upload per Round Trip Time Count*

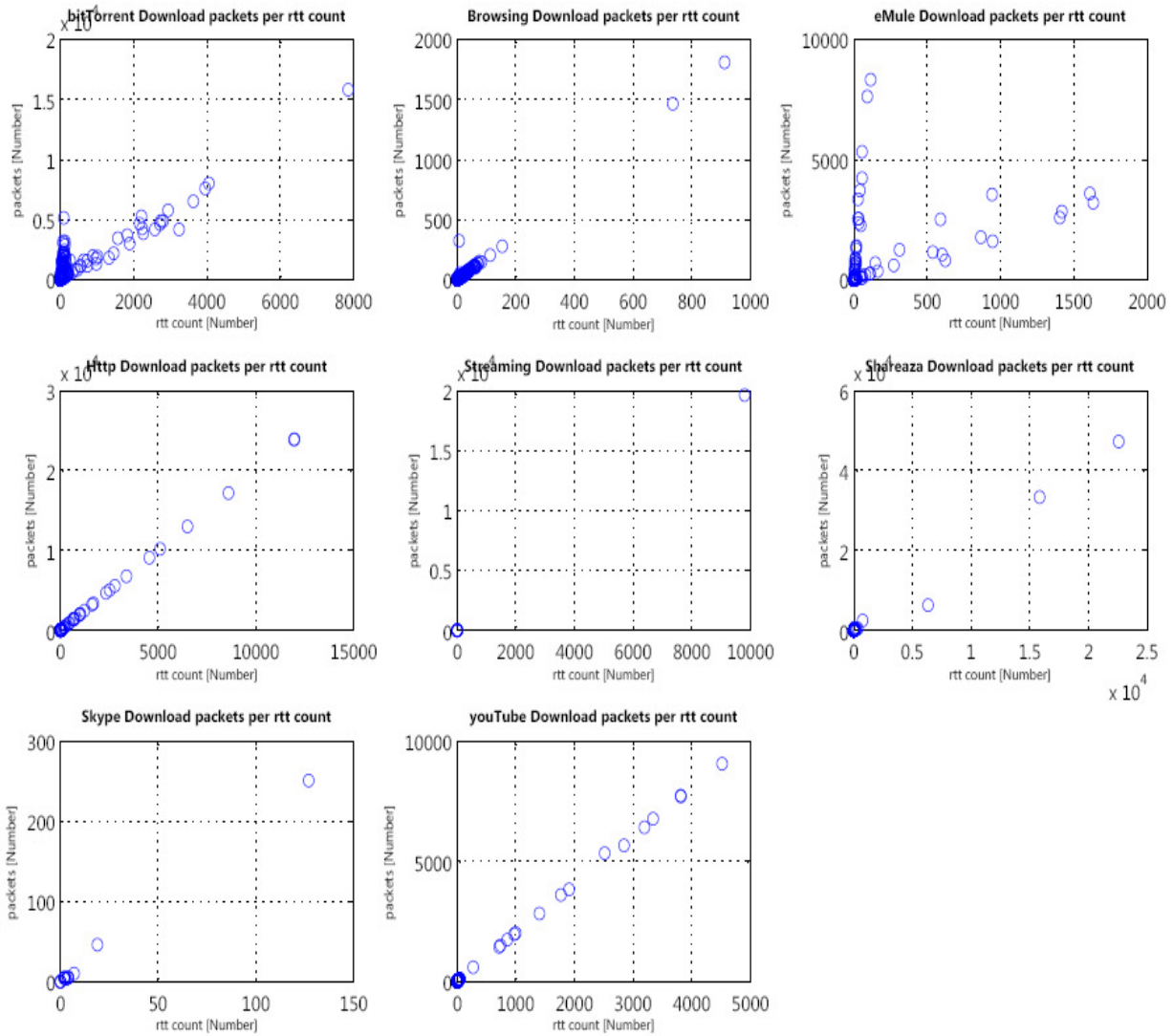

figura 14: *Packets Download per Round Trip Time Count*

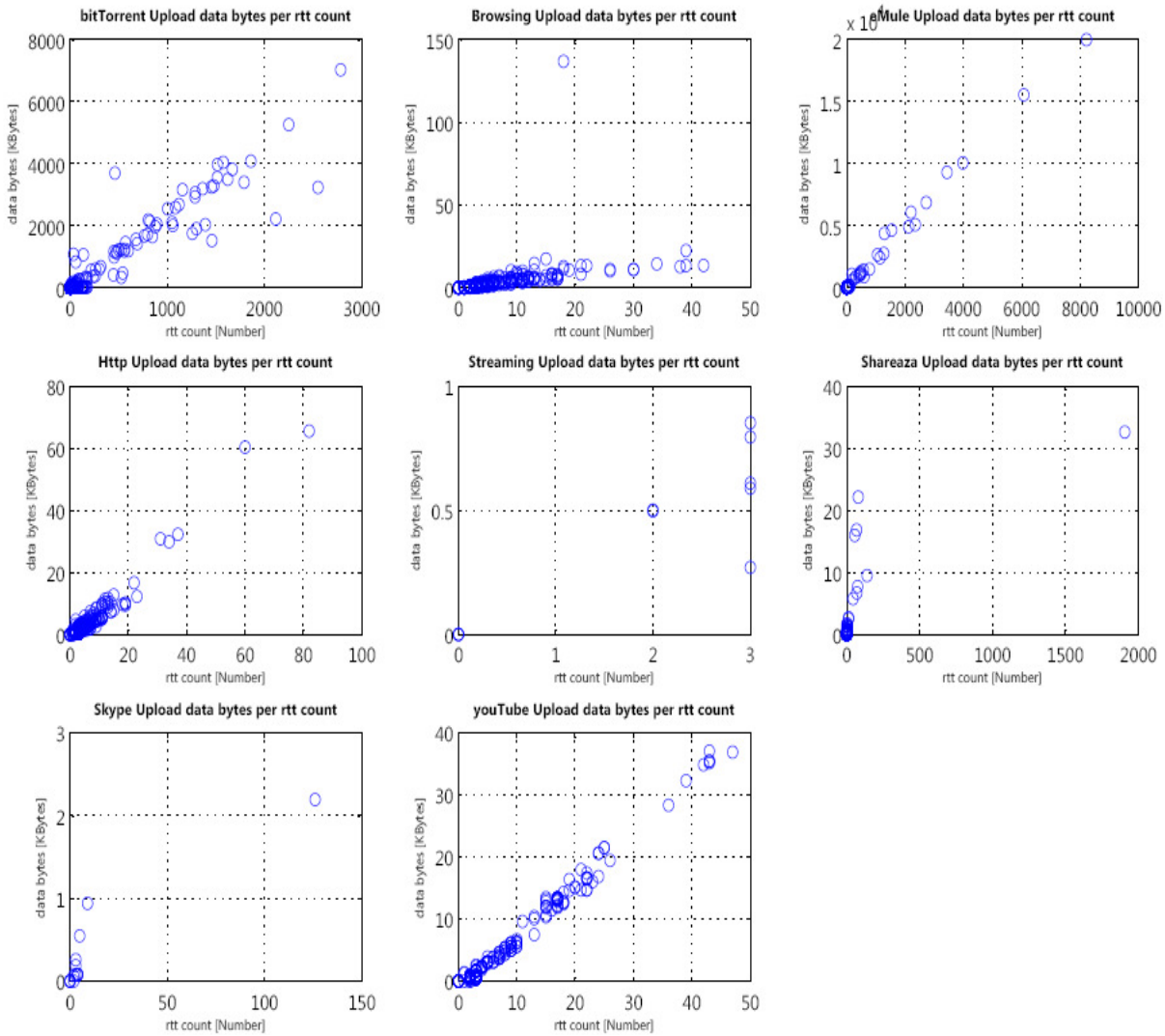

figura 15: *Data Bytes Upload per Round Trip Time Count*

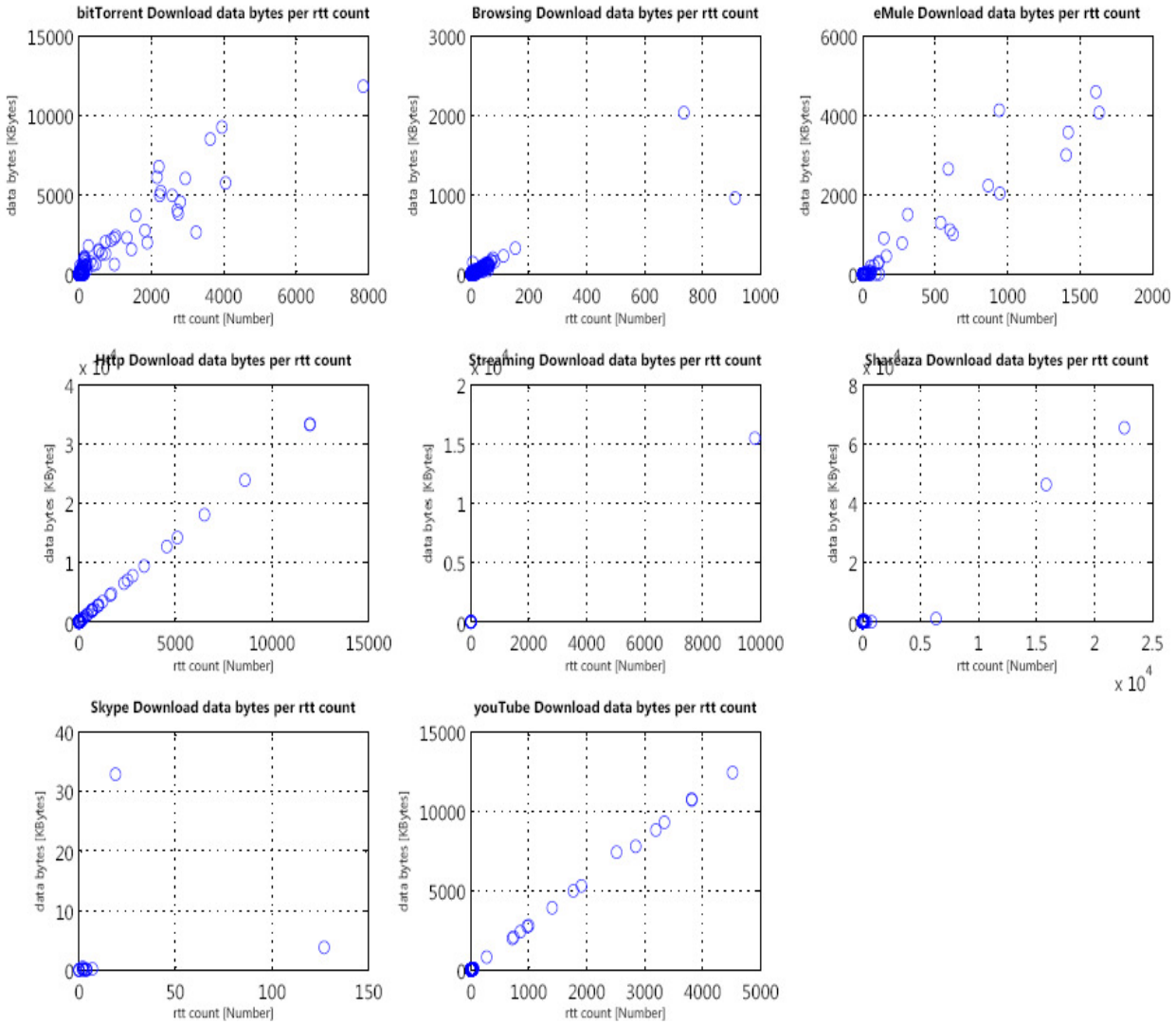

figura 16: *Data Bytes Download per Round Trip Time Count*

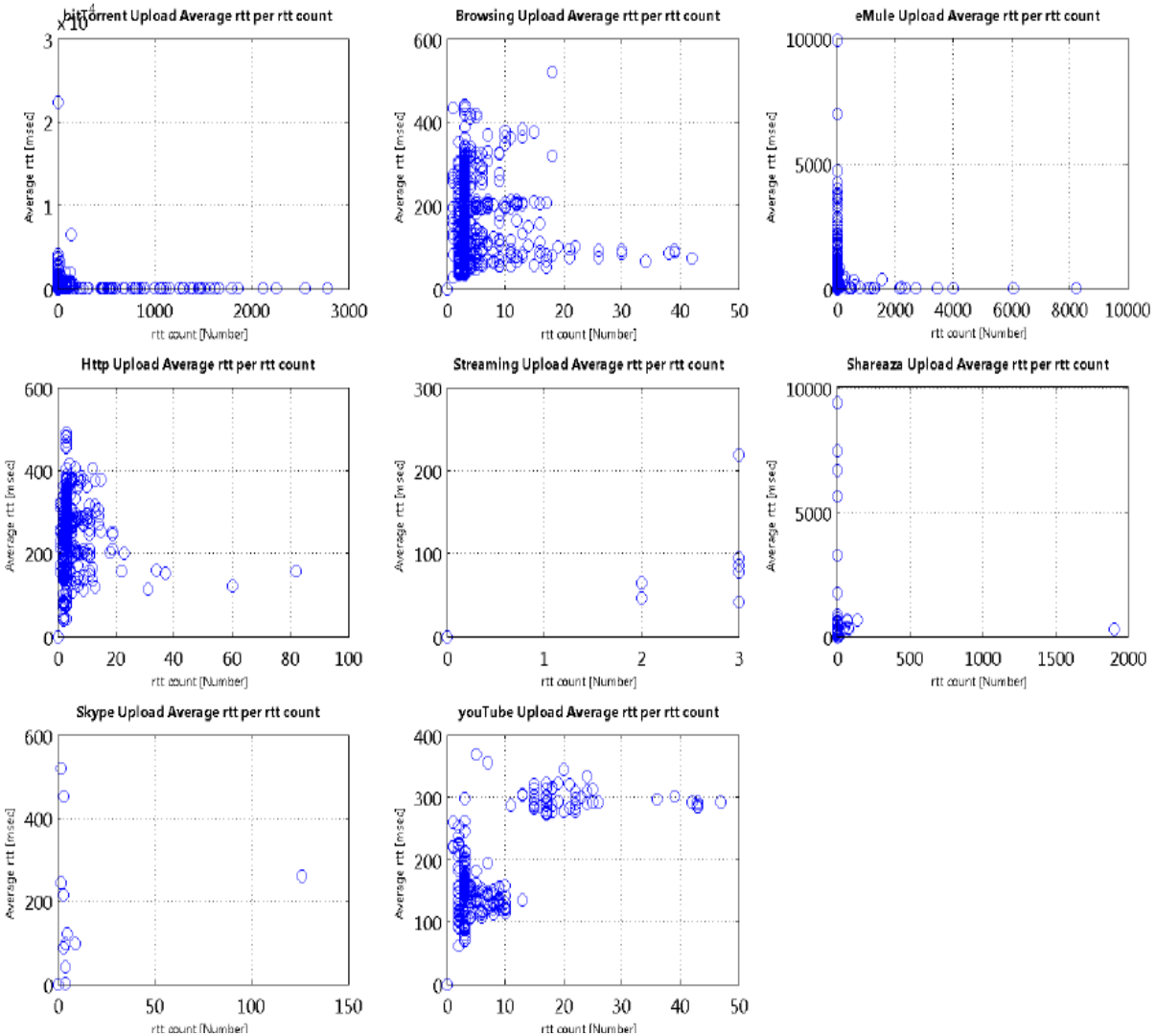

figura 17: *Average Round Trip Time Upload per Round Trip Time Count*

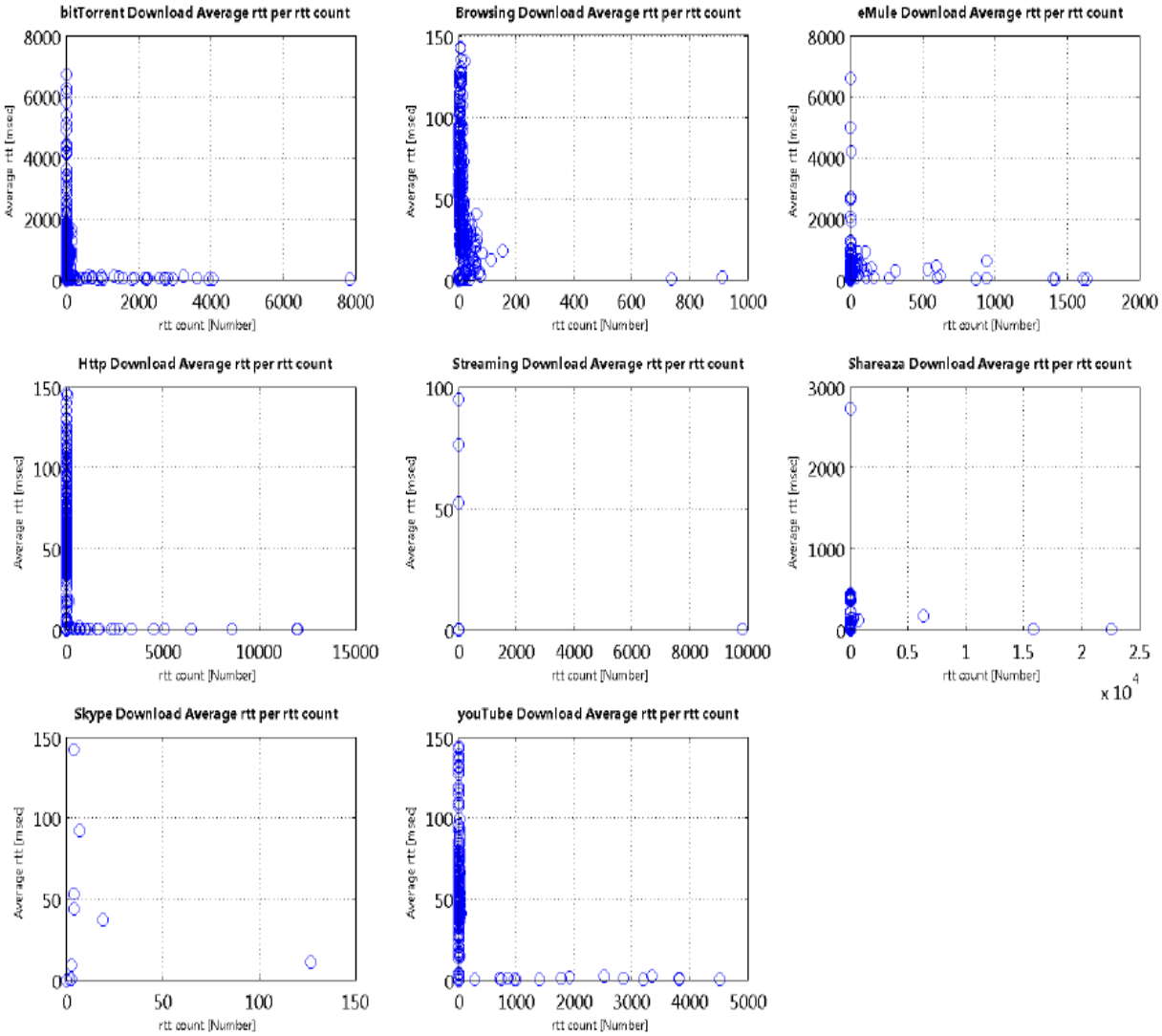

figura 18: *Average Round Trip Time Download per Round Trip Time Count*

Os gráficos anteriores permitem concluir que o número de contagens válidas de tempo entre o envio de uma mensagem e recepção da mensagem do tipo ACK correspondente é directamente proporcional à duração do fluxo, o que indicia que a contagem é feita de forma periódica.

É também possível verificar que os parâmetros relativos ao tempo médio entre o envio de uma mensagem e recepção da mensagem do tipo ACK correspondente são aqueles que, ao nível bidimensional, apresentam maiores diferenças entre os protocolos, sendo este um factor essencial para a caracterização pretendida.

5.4 CARACTERIZAÇÃO DE TRÁFEGO

O objectivo principal desta dissertação consiste na identificação e desenvolvimento de técnicas que permitam a caracterização do tráfego gerado por aplicações *P2P*.

O desenvolvimento do método de caracterização de tráfego foi realizado em MATLAB. Esta plataforma oferece um conjunto de funções base e o ambiente ideal para o tipo de processamento estatístico que se pretende.

O desenho das funções necessárias para a implementação de um método automático de caracterização de dados estatísticos obtidos a partir do *TSTAT*, levou à definição de 3 grandes grupos de funções:

- Identificação do conjunto de parâmetros de dados estatísticos que melhores resultados produz na Análise de Componentes Principais;
- Identificação das áreas de incidência de cada aplicação analisada;
- Validação de resultados.

5.4.1 ANÁLISE DE COMPONENTES PRINCIPAIS

Como vimos anteriormente, a Análise de Componentes Principais é um método quantitativo rigoroso que permite reduzir o número de parâmetros em conjuntos complexos, como é o caso dos devolvidos pelo *TSTAT*.

A função ***princomp*** é usada para o cálculo das componentes principais de um conjunto através do MATLAB.

Foi desenvolvida a função ***bestDataCombination*** que para um conjunto de dados estatísticos de entrada, calcula as componentes principais para cada combinação possível

dos parâmetros que compõem o conjunto. Para que deste processamento resulte uma diminuição no número de dimensões do conjunto é requisito que cada combinação tenha pelo menos 3 parâmetros. Para além disso, a função ***bestDataCombination*** devolve um relatório com a percentagem de variância relativa às duas primeiras componentes principais, identificando as combinações em que todas as aplicações atingem percentagens superiores a um limiar de entrada.

```
function pca_data = bestDataCombination (initial_data_sets, component, valid_pct, report_params)
```

INPUT

initial_data_sets – Estrutura indexada por um identificador numérico da aplicação. É composta por:

- **values** – Dados estatísticos da aplicação;
- **name** – Descrição identificativa da aplicação;

component – Estrutura composta pela descrição de cada parâmetro que compõe o conjunto de dados estatísticos das aplicações;

valid_pct – Limiar percentual para definição das combinações de parâmetros válidas (valor mínimo);

report_params – Estrutura composta por:

- **path** – Directoria de escrita do ficheiro do relatório;
- **filename** – Nome do ficheiro do relatório;
- **write** – Permite definir se o ficheiro deverá ser gerado (1) ou não (0).

OUTPUT

pca_data – Para além do relatório, a função devolve esta estrutura indexada por um identificador numérico da combinação. É composta por:

- **components_nbr** – Total de parâmetros dos dados estatísticos, que compõem a combinação;
- **valid_component_combination** – Identifica se a combinação é válida (1) ou não (0). Para que a combinação seja considerada válida é necessário que, para todas as aplicações, a percentagem associada às duas primeiras componentes principais seja superior ao limiar definido (**valid_pct**);
- **less_pct** – Valor da menor percentagem entre todas as aplicações, para esta combinação de parâmetros;
- **set** – Estrutura indexada por um identificador numérico da aplicação. Contém os resultados da Análise de Componentes Principais dos parâmetros da combinação, para cada aplicação.

A informação relativa a cada combinação possível é apresentada no relatório segundo o seguinte formato:

```
################################################################################
# Combination line:   l1 * Components: c1, c2, c3, ..., cn
#*******************************************************************************#
# app1: pct1%
# app2: pct2%
# ...
# appm: pctm%
# INVALID COMPONENT COMBINATION!
################################################################################
```

No final do relatório é apresentado um resumo dos resultados, com as percentagens mínimas observadas para cada uma das combinações de parâmetros.

```
#################################################################################
                COMBINATION   | COMPONENTS NBR |     MIN PCT
                      1       |         3      |     minpct1%
                      2       |         3      |     minpct2%
                     ...
                      k       |         n      |     minpctk%
```

De forma a identificar o *upload*, o *download*, ou ambos, foi feita a análise separada do tráfego gerado e recebido pela máquina observada.

Colocou-se como requisito para a escolha da combinação de parâmetros usada no restante processamento a percentagem mínima de 75% nas primeiras duas componentes principais de cada aplicação. Esta percentagem oferece a garantia de perda pouco significativa de informação relativamente ao conjunto de parâmetros inicial. De notar que a combinação de parâmetros encontrada passa a ser descrita em 75%, pelas primeiras duas componentes principais resultante da Análise de Componentes Principais.

A combinação com maior número de parâmetros que cumpriu com o requisito gerou o seguinte relatório:

UPLOAD

```
#################################################################################
# Combination line:   76 * Components: Completion time, rtt count, data bytes, ACK sent, packets
#*******************************************************************************#
# bitTorrent Upload: 93.0368%
# Browsing Upload: 75.1828%
# eMule Upload: 99.2863%
# Http Upload: 97.0731%
# Streaming Upload: 99.0978%
# Shareaza Upload: 91.7757%
# Skype Upload: 99.7684%
# youTube Upload: 89.8601%
#################################################################################
```

DOWNLOAD

```
#################################################################################
# Combination line:   76 * Components: Completion time, rtt count, data bytes, ACK sent, packets
#*******************************************************************************#
# bitTorrent Download: 96.5736%
# Browsing Download: 96.9390%
# eMule Download: 90.4392%
# Http Download: 99.9997%
# Streaming Download: 100.0000%
# Shareaza Download: 99.5915%
# Skype Download: 99.9253%
# youTube Download: 99.9947%
#################################################################################
```

5.4.2 IDENTIFICAÇÃO DE PADRÕES DE TRÁFEGO

Após a escolha da combinação de parâmetros a usar no restante processamento e respectiva Análise de Componentes Principais, podemos passar à identificação bidimensional de padrões de tráfego que distingam cada uma das aplicações.

Com este objectivo, foi desenvolvida a função MATLAB ***getIncidenceAreas*** que implementa o algoritmo que a seguir se descreve:

1. Determinação dos valores mínimos e máximos das duas primeiras componentes principais, entre todas as aplicações.

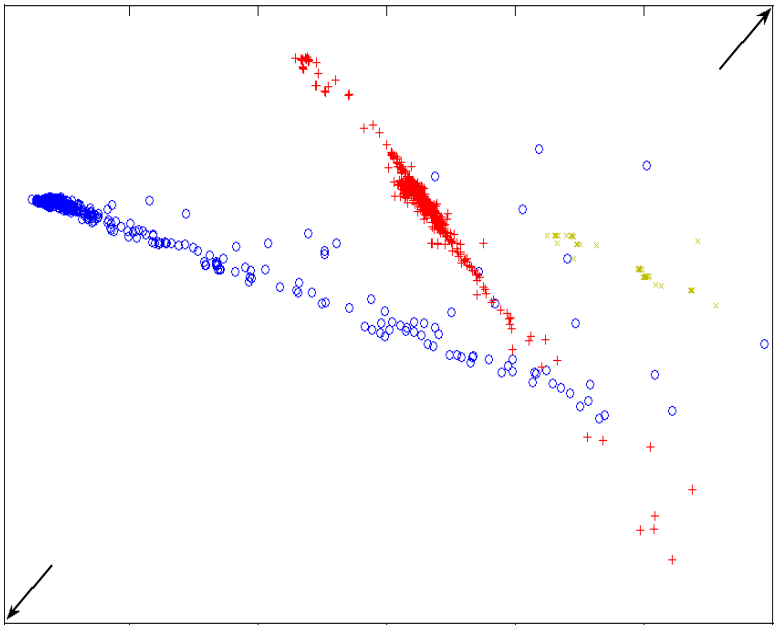

figura 19: Determinação dos valores mínimos e máximos

2. Divisão do espaço bidimensional obtido, num número de áreas rectangulares pré-estabelecido (parâmetro de entrada).

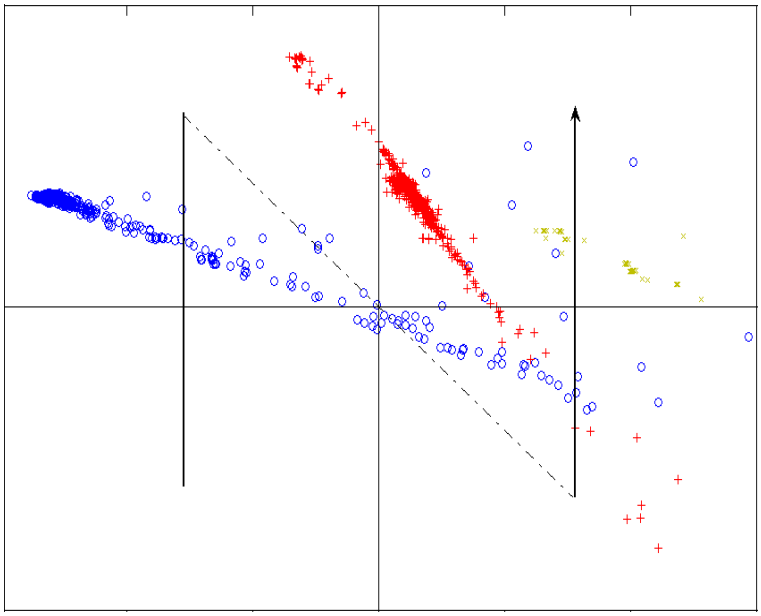

figura 20: Método de verificação das áreas

3. Para cada área é verificado se o número de pontos de uma aplicação, a que passaremos a designar por aplicação analisada, é significativamente superior ao das restantes aplicações. Esta operação é feita área a área, segundo a ordem definida na figura anterior:
    a. Áreas em que o número de pontos da aplicação analisada não é significativo (inferior a uma percentagem mínima pré-estabelecida) são automaticamente descartadas (consideradas inválidas).
    b. Áreas em que o número de pontos da aplicação analisada é significativo mas em que pelo menos uma aplicação tem um total de pontos não desprezável (superior a uma percentagem máxima pré-estabelecida), obriga à divisão da área pelo número pré-estabelecido de entrada e à repetição do processamento do ponto 3.

Durante o processamento de cada área, são assinaladas aquelas em que a aplicação analisada tem predominância de pontos sobre as restantes. A figura seguinte representa graficamente o resultado da aplicação do método descrito sobre um conjunto exemplo de dados estatísticos.

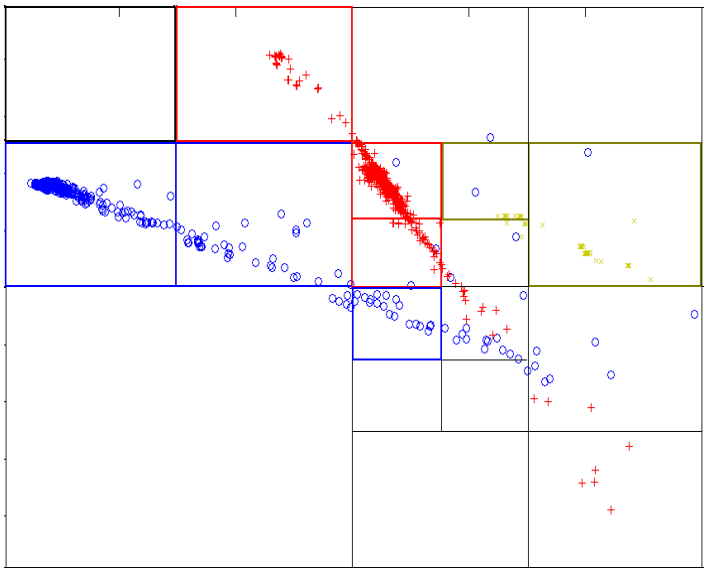

figura 21: Identificação das áreas relevantes de cada aplicação

A função dispõe do seguinte interface:

```
function [data_subsets, total_areas] = getIncidenceAreas (data_sets, analyzing_set_id,
nonsignificant_pct, relevant_pct, squares_nbr)
```

INPUT

data_sets – Contém a informação da combinação escolhida (presente na estrutura **set**): resultados da Análise de Componentes Principais dos parâmetros da combinação, para cada aplicação.

analyzing_set_id – Identificador numérico da aplicação a analisar;

nonsignificant_pct – Percentagem de pontos de uma aplicação, que não a aplicação analisada, a partir da qual uma área deverá ser considerada inválida;

relevant_pct – Percentagem mínima de pontos da aplicação analisada, para que a área seja considerada válida;

squares_nbr – Define o número de divisões que cada eixo de uma área deve sofrer, antes de ser reanalisada;

OUTPUT

data_subsets – Estrutura indexada por um identificador numérico da área. É composta por:

- **finished** – Parâmetro de controlo que permite à função marcar as áreas finalizadas (sem necessidade de nova divisão);
- **relevant_area** – Identifica se a área é válida (1) ou não (0);
- **min_values** – Contém os valores mínimos das duas primeiras componentes principais, que delimitam a área;
- **max_values** – Contém os valores máximos das duas primeiras componentes principais, que delimitam a área;
- **set** – Estrutura indexada por um identificador numérico da aplicação. Contém os seguintes parâmetros:
  - **values** – Dados estatísticos da combinação escolhida que se encontram na área (sub-conjunto do **data_sets**);
  - **occurrences** – Número de pontos contidos na área (tamanho de **values**);
  - **total_points** – Total de pontos do conjunto de dados estatísticos da aplicação;
  - **line** – Identificador das linhas do conjunto de dados estatísticos que se encontram na área.

total_areas – Número de áreas que compõem a estrutura **data_subsets**.

Esta função realiza o cálculo das áreas de interesse para uma aplicação. De forma a generalizar o processamento a todas as aplicações, foi desenvolvida a função ***incidenceAreasReport*** que gera um relatório com a informação relativa às áreas de interesse para cada aplicação. Neste relatório constam a percentagem de pontos encontrados na área para cada aplicação, bem como o valor percentual agregado para todas as áreas identificadas, por aplicação analisada. Esta informação é suficiente para verificar se o algoritmo usado permite identificar padrões no tráfego gerado pelas aplicações.

A função dispõe do seguinte interface:

```
function data_analysis = incidenceAreasReport (data_sets, nonsignificant_pct, relevant_pct,
squares_set, report_params)
```

INPUT

data_sets – Contém a informação da combinação escolhida (presente na estrutura **set**): resultados da Análise de Componentes Principais dos parâmetros da combinação, para cada aplicação.

nonsignificant_pct – Percentagem de pontos de uma aplicação, que não a aplicação analisada, a partir da qual uma área deverá ser considerada inválida;

relevant_pct – Percentagem mínima de pontos da aplicação analisada, para que a área seja considerada válida;

squares_set – Define um intervalo de valores usados como divisor de cada eixo de uma área, antes de ser reanalisada;

report_params – Estrutura composta por:

- **path** – Directoria de escrita do ficheiro do relatório;
- **filename** – Nome do ficheiro do relatório;
- **write** – Permite definir se o ficheiro deverá ser gerado (1) ou não (0).

OUTPUT

data_analysis – Estrutura indexada por um identificador numérico da aplicação. É composta por:

- **aproximation** – Estrutura indexada pelo divisor (**squares_set**) usado no cálculo das áreas. É composta por:
  - **data_subsets** – Estrutura indexada por um identificador numérico da área. É composta por:
    - **finished** – Parâmetro de controlo que permite à função marcar as áreas finalizadas (sem necessidade de nova divisão);
    - **relevant_area** – Identifica se a área é válida (1) ou não (0);
    - **min_values** – Contém os valores mínimos das duas primeiras componentes principais, que delimitam a área;
    - **max_values** – Contém os valores máximos das duas primeiras componentes principais, que delimitam a área;
    - **set** – Estrutura indexada por um identificador numérico da aplicação. Contém os seguintes parâmetros:
      - **values** – Dados estatísticos da combinação escolhida que se encontram na área (sub-conjunto do **data_sets**);
      - **occurrences** – Número de pontos contidos na área (tamanho de **values**);
      - **total_points** – Total de pontos do conjunto de dados estatísticos da aplicação;
      - **line** – Identificador das linhas do conjunto de dados estatísticos que se encontram na área.
  - **total_areas** – Número de áreas que compõem a estrutura **data_subsets**.

O relatório apresenta o seguinte formato:

```
################################################################################
# Incidence Areas Report at YYYY-MM-DD HH:MI:SS          Data Sets Nbr: setnbr#
################################################################################
* Non Significant Pct: min_pct%    *     Min Sqr Nbr  :   min_sqr
* Relevant Pct       : max_pct%    *     Max Sqr Nbr  :   max_sqr
---------------------------
  Set ID  |  Total Points
----------|----------------
   app1   |    total1
   app2   |    total2
...
   appn   |    totaln
---------------------------

################################################################################
# ANALYZING SET ID: appx                                  SQUARES NBR: sqry
################################################################################
****************
 AREA ID: areaz*  (min_x, min_y): (min_xz, min_yz)  (max_x, max_y): (max_xz, max_yz)
********************************************************************************
                                          | SetID |Occurrences|Contained Pct
------------------------------------------|-------|-----------|-------------
                                          |  app1 |   occur1  |  contpct1%
                                          |  app2 |   occur2  |  contpct2%
                                            ...
                                          |  appn |   occurn  |  contpctn%
------------------------------------------|-------|-----------|-------------
                          AGGREGATED VALUES|       |           |
                                          |  app1 |  aggocc1  |  aggcont1%
```

```
                                        |  app2  |  aggocc2  |  aggcont2%
                                        ...
                                        |  appn  |  aggoccn  |  aggcontn%
-------------------------------------------------------------------------
```

Sendo o objectivo principal da dissertação identificar padrões de tráfego para as aplicações *P2P* de partilha de ficheiros, usou-se os dados estatísticos destas aplicações como referência no estabelecimento das percentagens de entrada da função. Assim, como o número de fluxos registados nas capturas de cada uma destas aplicações é cerca de 1000, considerou-se requisito mínimo a existência de 1.5% de pontos (15) da aplicação analisada para que determinada área fosse considerada válida. Da mesma forma, áreas com mais de 1% de pontos (10) de qualquer outra aplicação foram consideradas inválidas (descartadas).

A função permite a definição de um conjunto de valores de divisão de áreas, para posterior comparação de resultados. No relatório usado como exemplo para o documento foram utilizados os valores inteiros compreendidos entre 2 e 5. Verificou-se que a resposta a valores superiores a 5 sofria uma degradação significativa, explicada pela redução imposta em cada divisão: áreas demasiado pequenas têm obrigatoriamente poucos pontos/percentagens baixas.

De seguida apresentam-se alguns dos resultados extraídos do relatório obtido:

UPLOAD

```
################################################################################
# INCIDENCE AREAS REPORT at 20-08-2007 21:27:33           Data Sets Nbr:   8   #
################################################################################
* Non Significant Pct: 1.00%    *     Min Sqr Nbr  :   2
* Relevant Pct       : 1.50%    *     Max Sqr Nbr  :   5
---------------------------
  Set ID  |  Total Points
----------|----------------
      1   |       1166
      2   |        824
      3   |        998
      4   |        697
      5   |         88
      6   |       1075
      7   |         51
      8   |        278
---------------------------
```

APPLICATION	OCCURRENCES	TOTAL (%)
Observed Application **BITTORRENT** (Squares Nbr: 3)		
bitTorrent	948	**81.30%**
Browsing	4	0.49%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%
Observed Application **BROWSING** (Squares Nbr: 5)		
bitTorrent	0	0.00%
Browsing	439	**53.28%**
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	1	0.36%
Observed Application **EMULE** (Squares Nbr: 4)		
bitTorrent	1	0.09%
Browsing	4	0.49%
eMule	876	**87.78%**
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%
Observed Application **HTTP** (Squares Nbr: 3)		
bitTorrent	0	0.00%
Browsing	0	0.00%
eMule	0	0.00%
HTTP	421	**60.40%**
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%

APPLICATION	OCCURRENCES	TOTAL (%)
Observed Application **STREAMING** (Squares Nbr: 3)		
bitTorrent	0	0.00%
Browsing	1	0.12%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	81	**92.05%**
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%
Observed Application **SHAREAZA** (Squares Nbr: 4)		
bitTorrent	0	0.00%
Browsing	1	0.12%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	1001	**93.12%**
Skype	0	0.00%
youTube	0	0.00%
Observed Application **SKYPE** (Squares Nbr: 4)		
bitTorrent	0	0.00%
Browsing	0	0.00%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	40	**100.00%**
youTube	0	0.00%
Observed Application **YOUTUBE** (Squares Nbr: 2)		
bitTorrent	3	0.26%
Browsing	1	0.12%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	133	**47.84%**

Tabela 5: Total de pontos contidos nas áreas (*Upload*)

DOWNLOAD

```
################################################################################
# INCIDENCE AREAS REPORT at 20-08-2007 21:41:10          Data Sets Nbr:   8   #
################################################################################
* Non Significant Pct: 1.00%     *     Min Sqr Nbr  :   2
* Relevant Pct       : 1.50%     *     Max Sqr Nbr  :   5
---------------------------
  Set ID   |  Total Points
-----------|---------------
      1    |        1166
      2    |         824
      3    |         998
      4    |         697
      5    |          88
      6    |        1075
      7    |          51
      8    |         278
```

APPLICATION	OCCURRENCES	TOTAL (%)
Observed Application **BITTORRENT** (Squares Nbr: 2)		
bitTorrent	908	**77.87%**
Browsing	0	0.00%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	1	0.36%
Observed Application **BROWSING** (Squares Nbr: 2)		
bitTorrent	0	0.00%
Browsing	384	**46.60%**
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	3	1.08%
Observed Application **EMULE** (Squares Nbr: 3)		
bitTorrent	3	0.26%
Browsing	2	0.24%
eMule	822	**82.36%**
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%

APPLICATION	OCCURRENCES	TOTAL (%)
Observed Application **STREAMING** (Squares Nbr: 2)		
bitTorrent	0	0.00%
Browsing	1	0.12%
eMule	5	0.50%
HTTP	3	0.43%
Streaming	85	**96.59%**
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%
Observed Application **SHAREAZA** (Squares Nbr: 2)		
bitTorrent	0	0.00%
Browsing	0	0.00%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	1022	**95.07%**
Skype	0	0.00%
youTube	0	0.00%
Observed Application **SKYPE** (Squares Nbr: 3)		
bitTorrent	0	0.00%
Browsing	0	0.00%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	39	**100.00%**
youTube	0	0.00%

Observed Application HTTP (Squares Nbr: 2)		
bitTorrent	4	0.34%
Browsing	1	0.12%
eMule	0	0.00%
HTTP	498	**71.45%**
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%

Observed Application YOUTUBE (Squares Nbr: 2)		
bitTorrent	1	0.09%
Browsing	0	0.00%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	129	**46.40%**

Tabela 6: Total de pontos contidos nas áreas (*Download*)

A função desenvolvida permite a identificação de áreas de predominância de cada aplicação analisada. Em particular para as aplicações *P2P* de partilha de ficheiros, foram identificadas áreas de tráfego (*upload/download*) que circunscrevem mais de 75% dos seus pontos. Para além disso, as restantes aplicações não têm qualquer expressão nessas áreas: menos de 1% de pontos presentes.

O número de áreas é dependente do tipo de tráfego (*upload/download*), do número de fluxos e da própria aplicação. Para os ensaios realizados são precisas menos de 10 áreas de predominância distintas por aplicação *P2P* de partilha de ficheiros para identificar mais de 75% dos seus pontos.

Estes resultados oferecem um grau elevado de certeza quanto à capacidade de caracterização do tráfego gerado por este tipo de aplicações. No ponto seguinte, encontra-se descrito o processo usado para a validação destes resultados.

5.4.3 VALIDAÇÃO DOS RESULTADOS

Para a verificação da fiabilidade do algoritmo implementado, foi realizado o processamento descrito em 5.4.1 e 5.4.2 para 2 sub-conjuntos dos dados estatísticos originais.

Apesar da relação existente, os 3 conjuntos resultantes (original e 2 sub-conjuntos) têm valores e vectores próprios distintos, como descrito no capítulo 4. Refira-se ainda que a divisão diminui o número de fluxos, reduzindo por isso a informação caracterizante do tráfego gerado pela aplicação. Estes factos implicam obrigatoriamente valores das componentes principais diferentes e consequentemente áreas de predominância de

tráfego distintas. Contudo, como veremos de seguida, as características do tráfego são mantidas, ocorrendo uma translação dos seus pontos.

Este processo permitiu validar que o método desenvolvido é eficiente na verificação da possibilidade de caracterização do tráfego gerado por uma aplicação. Para além disso, foi possível verificar que cada uma das aplicações *P2P* de partilha de ficheiros analisadas (*bitTorrent*, *eMule* e *Shareaza*), geram tráfego com características que as distingue de qualquer outra aplicação.

Para a combinação de parâmetros usada em 5.4.1 registaram-se os seguintes resultados:

UPLOAD

Sub-Conjunto 1

```
################################################################################
# Combination line:   76 * Components: Completion time, rtt count, data bytes, ACK sent, packets
#*******************************************************************************#
# bitTorrent Upload: 86.6337%
# Browsing Upload: 74.2623%
# eMule Upload: 99.2670%
# Http Upload: 94.8596%
# Streaming Upload: 95.5003%
# Shareaza Upload: 99.5257%
# Skype Upload: 95.6619%
# youTube Upload: 91.1946%
# INVALID COMPONENT COMBINATION!
################################################################################
```

Sub-Conjunto 2

```
################################################################################
# Combination line:   76 * Components: Completion time, rtt count, data bytes, ACK sent, packets
#*******************************************************************************#
# bitTorrent Upload: 96.3150%
# Browsing Upload: 92.6834%
# eMule Upload: 99.8833%
# Http Upload: 97.6976%
# Streaming Upload: 99.9997%
# Shareaza Upload: 96.8254%
# Skype Upload: 99.9999%
# youTube Upload: 88.8535%
################################################################################
```

DOWNLOAD

Sub-Conjunto 1

```
################################################################################
# Combination line:   76 * Components: Completion time, rtt count, data bytes, ACK sent, packets
#*******************************************************************************#
# bitTorrent Download: 98.1044%
# Browsing Download: 96.8964%
# eMule Download: 93.3577%
# Http Download: 99.9997%
# Streaming Download: 94.6996%
# Shareaza Download: 95.4520%
# Skype Download: 97.5220%
# youTube Download: 99.9887%
################################################################################
```

Sub-Conjunto 2

```
################################################################################
# Combination line:   76 * Components: Completion time, rtt count, data bytes, ACK sent, packets
#*******************************************************************************#
# bitTorrent Download: 96.2966%
# Browsing Download: 98.4415%
# eMule Download: 90.2737%
# Http Download: 99.9998%
# Streaming Download: 100.0000%
# Shareaza Download: 99.6104%
# Skype Download: 100.0000%
# youTube Download: 99.9981%
################################################################################
```

As primeiras duas componentes principais do tráfego relativo a *Browsing* do sub-conjunto 1 são responsáveis por uma percentagem da variância dos dados estatísticos ligeiramente abaixo dos 75% (percentagem mínima estipulada). Contudo, para as aplicações *P2P* de partilha de ficheiros este valor percentual ultrapassa largamente o requisito referido, pelo que a diferença foi ignorada. Para mais, como o objectivo final é a comparação destes resultados com os obtidos para o conjunto principal, é imperativo a utilização da mesma combinação de parâmetros estatísticos.

Com os resultados da Análise de Componente Principais para a combinação de parâmetros *TSTAT* referida, geraram-se os relatórios das áreas relativas aos 2 sub-conjuntos, dos quais se extraíram os seguintes valores:

UPLOAD

Sub-Conjunto 1

```
################################################################################
# INCIDENCE AREAS REPORT at 07-10-2007 16:41:16          Data Sets Nbr:   8   #
################################################################################
* Non Significant Pct: 1.00%    *     Min Sqr Nbr  :   2
* Relevant Pct       : 1.50%    *     Max Sqr Nbr  :   5
----------------------------
  Set ID   |  Total Points
-----------|----------------
      1    |         583
      2    |         412
      3    |         499
      4    |         349
      5    |          44
```

```
6  |  538
7  |  26
8  |  139
```

APPLICATION	OCCURRENCES	TOTAL (%)
Observed Application **BITTORRENT** (Squares Nbr: 5)		
bitTorrent	462	**79.25%**
Browsing	4	0.97%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%
Observed Application **BROWSING** (Squares Nbr: 2)		
bitTorrent	0	0.00%
Browsing	439	**52.18%**
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	1	0.72%
Observed Application **EMULE** (Squares Nbr: 2)		
bitTorrent	1	0.17%
Browsing	3	0.73%
eMule	405	**81.16%**
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%
Observed Application **HTTP** (Squares Nbr: 5)		
bitTorrent	0	0.00%
Browsing	0	0.00%
eMule	0	0.00%
HTTP	218	**62.46%**
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%

APPLICATION	OCCURRENCES	TOTAL (%)
Observed Application **STREAMING** (Squares Nbr: 2)		
bitTorrent	0	0.00%
Browsing	0	0.00%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	44	**100.00%**
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%
Observed Application **SHAREAZA** (Squares Nbr: 2)		
bitTorrent	2	0.34%
Browsing	1	0.24%
eMule	0	0.00%
HTTP	1	0.29%
Streaming	0	0.00%
Shareaza	502	**93.31%**
Skype	0	0.00%
youTube	0	0.00%
Observed Application **SKYPE** (Squares Nbr: 3)		
bitTorrent	0	0.00%
Browsing	0	0.00%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	26	**100.00%**
youTube	0	0.00%
Observed Application **YOUTUBE** (Squares Nbr: 2)		
bitTorrent	0	0.00%
Browsing	0	0.00%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	73	**52.52%**

Tabela 7: Total de pontos contidos nas áreas (*Upload* Sub-Conjunto 1)

Sub-Conjunto 2

```
################################################################################
# INCIDENCE AREAS REPORT at 07-10-2007 16:45:46          Data Sets Nbr:   8   #
################################################################################
* Non Significant Pct: 1.00%     *     Min Sqr Nbr  :   2
* Relevant Pct       : 1.50%     *     Max Sqr Nbr  :   5
---------------------------
  Set ID   |  Total Points
-----------|---------------
      1    |         583
      2    |         412
      3    |         499
      4    |         348
      5    |          44
      6    |         537
      7    |          25
      8    |         139
---------------------------
```

APPLICATION	OCCURRENCES	TOTAL (%)
Observed Application **BITTORRENT** (Squares Nbr: 3)		
bitTorrent	487	**83.53%**
Browsing	0	0.00%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%
Observed Application **BROWSING** (Squares Nbr: 3)		
bitTorrent	1	0.17%
Browsing	228	**55.34%**
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	1	0.72%
Observed Application **EMULE** (Squares Nbr: 3)		
bitTorrent	0	0.00%
Browsing	0	0.00%
eMule	463	**92.79%**
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%

APPLICATION	OCCURRENCES	TOTAL (%)
Observed Application **STREAMING** (Squares Nbr: 3)		
bitTorrent	0	0.00%
Browsing	0	0.00%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	44	**100.00%**
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%
Observed Application **SHAREAZA** (Squares Nbr: 5)		
bitTorrent	0	0.00%
Browsing	0	0.00%
eMule	1	0.20%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	513	**95.53%**
Skype	0	0.00%
youTube	0	0.00%
Observed Application **SKYPE** (Squares Nbr: 4)		
bitTorrent	0	0.00%
Browsing	0	0.00%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	25	**100.00%**
youTube	0	0.00%

Observed Application HTTP (Squares Nbr: 5)		
bitTorrent	0	0.00%
Browsing	2	0.49%
eMule	0	0.00%
HTTP	261	**75.00%**
Streaming	0	0.00%
Shareaza	3	0.56%
Skype	0	0.00%
youTube	1	0.72%

Observed Application YOUTUBE (Squares Nbr: 3)		
bitTorrent	4	0.69%
Browsing	8	1.94%
eMule	2	0.40%
HTTP	2	0.57%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	102	**73.38%**

Tabela 8: Total de pontos contidos nas áreas (*Upload* Sub-Conjunto 2)

DOWNLOAD

Sub-Conjunto 1

```
################################################################################
# INCIDENCE AREAS REPORT at 07-10-2007 19:45:40          Data Sets Nbr:   8   #
################################################################################
* Non Significant Pct: 1.00%    *     Min Sqr Nbr  :   2
* Relevant Pct       : 1.50%    *     Max Sqr Nbr  :   5
----------------------------
  Set ID   |  Total Points
-----------|----------------
      1    |         583
      2    |         412
      3    |         499
      4    |         349
      5    |          44
      6    |         538
      7    |          26
      8    |         139
----------------------------
```

APPLICATION	OCCURRENCES	TOTAL (%)
Observed Application BITTORRENT (Squares Nbr: 2)		
bitTorrent	407	**69.81%**
Browsing	2	0.49%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	2	1.44%

APPLICATION	OCCURRENCES	TOTAL (%)
Observed Application STREAMING (Squares Nbr: 4)		
bitTorrent	0	0.00%
Browsing	1	0.24%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	44	**100.00%**
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%

Observed Application BROWSING (Squares Nbr: 5)		
bitTorrent	0	0.00%
Browsing	221	**53.64%**
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%

Observed Application SHAREAZA (Squares Nbr: 3)		
bitTorrent	0	0.00%
Browsing	0	0.00%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	507	**94.24%**
Skype	0	0.00%
youTube	0	0.00%

Observed Application EMULE (Squares Nbr: 2)		
bitTorrent	1	0.17%
Browsing	3	0.73%
eMule	423	**84.77%**
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%

Observed Application SKYPE (Squares Nbr: 3)		
bitTorrent	0	0.00%
Browsing	0	0.00%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	26	**100.00%**
youTube	0	0.00%

Observed Application HTTP (Squares Nbr: 2)		
bitTorrent	8	1.37%
Browsing	1	0.24%
eMule	8	1.60%
HTTP	261	**74.79%**
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%

Observed Application YOUTUBE (Squares Nbr: 2)		
bitTorrent	0	0.00%
Browsing	0	0.00%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	75	**53.96%**

Tabela 9: Total de pontos contidos nas áreas (*download* Sub-Conjunto 1)

Sub-Conjunto 2

```
################################################################################
# INCIDENCE AREAS REPORT at 07-10-2007 20:17:34          Data Sets Nbr:   8   #
################################################################################
* Non Significant Pct: 1.00%    *     Min Sqr Nbr  :   2
* Relevant Pct       : 1.50%    *     Max Sqr Nbr  :   5
----------------------------
  Set ID   |  Total Points
-----------|----------------
      1    |         583
      2    |         412
      3    |         499
      4    |         348
      5    |          44
      6    |         537
      7    |          25
      8    |         139
----------------------------
```

APPLICATION	OCCURRENCES	TOTAL (%)
Observed Application **BITTORRENT** (Squares Nbr: 2)		
bitTorrent	476	**81.65%**
Browsing	0	0.00%
eMule	4	0.80%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%
Observed Application **BROWSING** (Squares Nbr: 2)		
bitTorrent	0	0.00%
Browsing	244	**59.22%**
eMule	1	0.20%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%
Observed Application **EMULE** (Squares Nbr: 2)		
bitTorrent	0	0.00%
Browsing	1	0.24%
eMule	425	**85.17%**
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%
Observed Application **HTTP** (Squares Nbr: 5)		
bitTorrent	0	0.00%
Browsing	3	0.73%
eMule	0	0.00%
HTTP	298	**85.63%**
Streaming	0	0.00%
Shareaza	1	0.19%
Skype	0	0.00%
youTube	0	0.00%

APPLICATION	OCCURRENCES	TOTAL (%)
Observed Application **STREAMING** (Squares Nbr: 2)		
bitTorrent	0	0.00%
Browsing	0	0.00%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	44	**100.00%**
Shareaza	0	0.00%
Skype	0	0.00%
youTube	0	0.00%
Observed Application **SHAREAZA** (Squares Nbr: 4)		
bitTorrent	0	0.00%
Browsing	0	0.00%
eMule	1	0.20%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	515	**95.90%**
Skype	0	0.00%
youTube	0	0.00%
Observed Application **SKYPE** (Squares Nbr: 2)		
bitTorrent	0	0.00%
Browsing	0	0.00%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	25	**100.00%**
youTube	0	0.00%
Observed Application **YOUTUBE** (Squares Nbr: 2)		
bitTorrent	0	0.00%
Browsing	0	0.00%
eMule	0	0.00%
HTTP	0	0.00%
Streaming	0	0.00%
Shareaza	0	0.00%
Skype	0	0.00%
youTube	74	**53.24%**

Tabela 10: Total de pontos contidos nas áreas (*download* Sub-Conjunto 2)

Os valores apresentados mostram novamente a eficácia do algoritmo na caracterização das áreas de predominância das aplicações *P2P* de partilha de ficheiros. Apesar de existir

um valor manifestamente abaixo do expectável (≈ 70%), ainda assim é suficientemente elevado para a garantia da manutenção das características do tráfego original.

A representação gráfica das duas primeiras componentes principais dos conjuntos (principal e 2 sub-conjuntos) apresentada no ANEXO C, permite aferir visualmente a similaridade de características entre os dados que os compõem.

6 CONCLUSÕES

O aumento acentuado no volume de tráfego a que se tem assistido nos últimos anos deve-se, em grande parte, ao aparecimento das aplicações *P2P*. Para se adaptarem as actuais infra-estruturas de *hardware* e *software* de forma a melhorar o serviço prestado é importante, em primeiro lugar, conseguir identificar os principais culpados na degradação de *QoS*. Nos ensaios realizados foi possível verificar a utilização elevada de largura de banda por parte destas aplicações, como de resto revelaram os resultados das capturas de tráfego.

Apesar dos condicionalismos apontados, as aplicações *P2P* são responsáveis por uma das maiores revoluções da Internet, após a sua criação. Actualmente, são inúmeros os serviços disponibilizados através de protocolos *P2P*, em áreas tão diversas como o entretenimento, a educação e a comunicação, entre outras.

Como referido anteriormente no documento, as aplicações *P2P* de partilha de ficheiros enfrentam complexas questões legais devido à disponibilização sem autorização de conteúdos com direitos de autor. Este é um forte entrave à realização de estudos sobre este tipo de aplicações, principalmente no meio académico. Será também por isso que ainda não existem formas eficazes de combater a troca destes conteúdos, assim como a limitação na utilização de largura de banda por parte destas aplicações.

O desenvolvimento de técnicas que permitam a identificação do tráfego gerado por aplicações *P2P* de partilha de ficheiros, quando inserido no tráfego Internet global, assume um relevo importantíssimo para a construção de métodos eficientes de controlo deste tipo de aplicações. Os dados estatísticos recolhidos nos ensaios são por si só uma fonte inesgotável de informação, possibilitando novas investigações/projectos.

O estudo da arquitectura das aplicações *P2P* de partilha de ficheiros permitiu:

- Identificar os principais tipos de mensagens e arquitectura de rede usadas;
- O desenho e execução dos cenários de teste;
- Compreender os resultados obtidos e facilitar as tomadas de decisão nas diferentes fases do trabalho.

A escolha de aplicações populares e/ou geradoras de volume de tráfego elevado teve por objectivo simular, de uma forma o mais realista possível, o tráfego que podemos encontrar na Internet. Para além disso, a diversidade de protocolos e tipos de aplicações usadas é sinónimo de tráfego com características variadas, aumentando o grau de confiança nos resultados obtidos.

A realização de ensaios semelhantes para as diferentes aplicações de partilha de ficheiros permitiu não só obter resultados válidos para a análise pretendida mas também comparar o comportamento de cada uma delas. No que diz respeito a aplicações *P2P*, a aplicação *bitTorrent* mostrou o melhor desempenho, como de resto expressam os valores de tráfego apresentados na Tabela 2 e Tabela 3: apresentou não só a velocidade de transferência mais elevada, e que é indissociável do maior número de fontes por conteúdo pretendido, mas também o menor tempo de espera para o início de transferência. Outra das características do *bitTorrent* é o facto do número de fluxos com mensagens sem *payload* (sincronização, *acknowledge*, etc.) ser significativamente superior a qualquer outra aplicação *P2P* de partilha de ficheiros.

A representação gráfica de um conjunto de dados permite identificar visualmente as suas principais características e possibilita a comparação com outros conjuntos. Contudo, a representação de conjuntos de dados com mais de dois parâmetros é de interpretação complexa ou impossível. A Análise de Componentes Principais abrevia esta dificuldade, concentrando a informação de um conjunto de múltiplos parâmetros nas primeiras duas componentes principais resultantes. Após a aplicação do método *PCA*, é possível a representação de qualquer conjunto de dados sem grande perda de informação, comparativamente com o conjunto original.

O método de caracterização desenvolvido no âmbito da dissertação permitiu mostrar que é possível identificar características marcantes do tráfego gerado por aplicações *P2P* de partilha de ficheiros quando inserido no tráfego Internet global:

- Em primeiro lugar, a Análise de Componentes Principais ao conjunto de dados resultante das capturas realizadas nos ensaios possibilitou a redução da sua dimensionalidade, sem perda significativa de informação. Este facto foi confirmado pela elevada percentagem de variância presente nas duas primeiras componentes principais do conjunto.

- A capacidade de identificação de percentagens elevadas de pontos de uma dada aplicação em áreas em que as restantes aplicações não têm expressão, associado ao facto da Análise de Componentes Principais garantir a ortogonalidade dos conjuntos, mostra que as aplicações têm características de tráfego únicas e que é possível a sua identificação num meio com múltiplos tipos de tráfego. Com as áreas de predominância da aplicação é possível reconstruir o conjunto de dados original à custa da informação que lhe confere carácter único, comparativamente às restantes aplicações.

Como comentário final, refira-se que a pesquisa bibliográfica permitiu aferir do estado da arte relativamente à investigação feita sobre protocolos *P2P*. É notório o interesse sobre a matéria; contudo o constante aparecimento e desaparecimento de aplicações/protocolos *P2P*, associado às complicações legais que os envolvem dificulta em muito a sua análise e compreensão. A principal contribuição desta dissertação foi a implementação de um método que permite a caracterização do tráfego gerado por qualquer aplicação. Em particular, no que diz respeito às principais aplicações *P2P* de partilha de ficheiros, foi possível verificar que o seu tráfego contém características que lhe confere unicidade. Esta informação é essencial para a definição de futuras investigações.

7 TRABALHO FUTURO

A dissertação revelou um conjunto de conclusões importantes no desenvolvimento de técnicas de caracterização de tráfego. Este é o início de um trabalho que merece ser continuado na tentativa da identificação de formas de combate à distribuição ilegal de conteúdos com direitos de autor e na melhoria contínua do serviço prestado pelos *ISP*s.

Neste domínio, gostaria de destacar a importância do estudo do *bitTorrent* que, para além de ser o protocolo *P2P* de partilha de ficheiros com maior número de utilizadores é o responsável por mais de metade do tráfego Internet da actualidade.

Como referido anteriormente, o método desenvolvido permitiu verificar que a caracterização de tráfego é possível. Como complemento a este método sugiro o desenvolvimento de um método que permita a identificação das aplicações responsáveis pelo tráfego existente numa qualquer captura realizada, através da análise das características identificadas para essas mesmas aplicações.

ANEXO A

A função MATLAB ***applicationPlots*** foi desenvolvida para a geração de gráficos bidimensionais de conjuntos de dados relativos a múltiplas aplicações.

```
function applicationPlots (data_sets, components)
components_size = size(components);
data_sets_size = size(data_sets);
for y = 1:components_size(1,2)
    for x = 1:components_size(1,2)
        if y < x
            figure;
            for data_set_id = 1:data_sets_size(1,2)
                subplot(3,3,data_set_id);
                plot(data_sets(data_set_id).values(:,x)/components(x).factor,
data_sets(data_set_id).values(:,y)/components(y).factor, 'ob');
                xlabel([components(x).name ' [' components(x).unit ']'], 'FontSize', 7);
                ylabel([components(y).name ' [' components(y).unit ']'], 'FontSize', 7);
                title([data_sets(data_set_id).name ' ' components(y).name ' per '
components(x).name], 'FontWeight', 'bold', 'FontSize', 8);
                grid on;
            end;
        end;
    end;
end;
```

A estrutura **`data_sets`** contém os dados estatísticos das aplicações a analisar, enquanto que **`components`** define o grupo de parâmetros objecto de análise da função.

Os gráficos considerados mais relevantes para a combinação de parâmetros escolhida encontram-se no ponto 5.3, sendo os restantes apresentados de seguida.

TEMPO MÉDIO DE *ROUND TRIP*

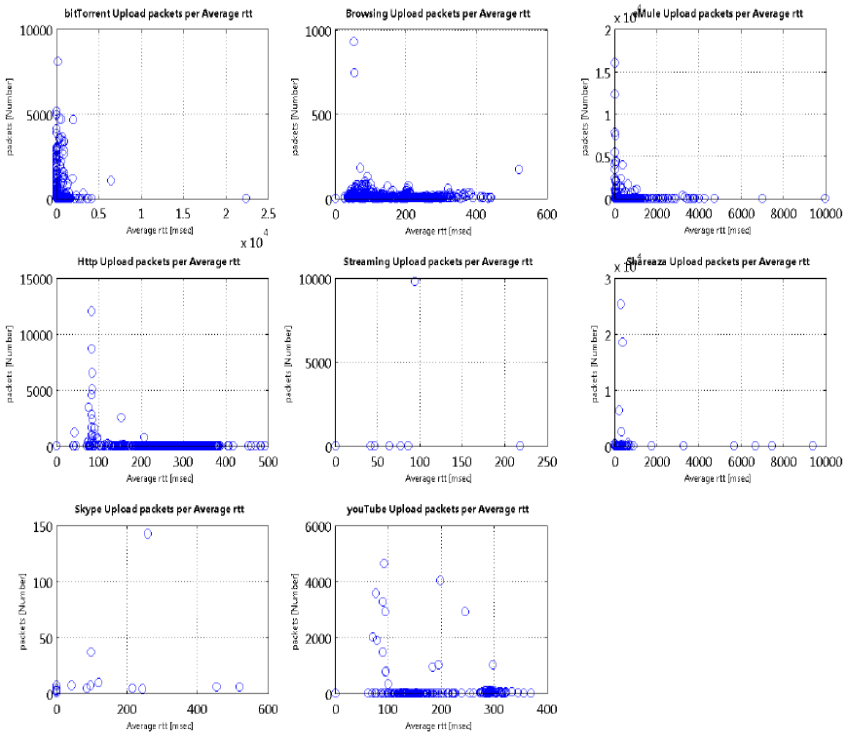

figura 22: *Packets Upload per Average Round Trip Time*

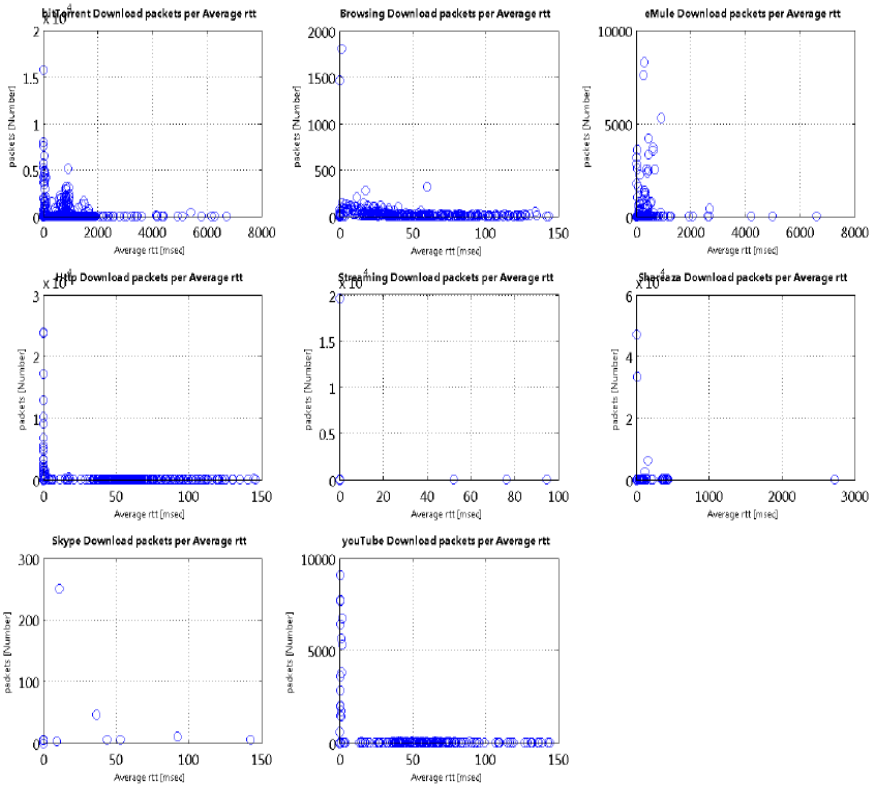

figura 23: *Packets Download per Average Round Trip Time*

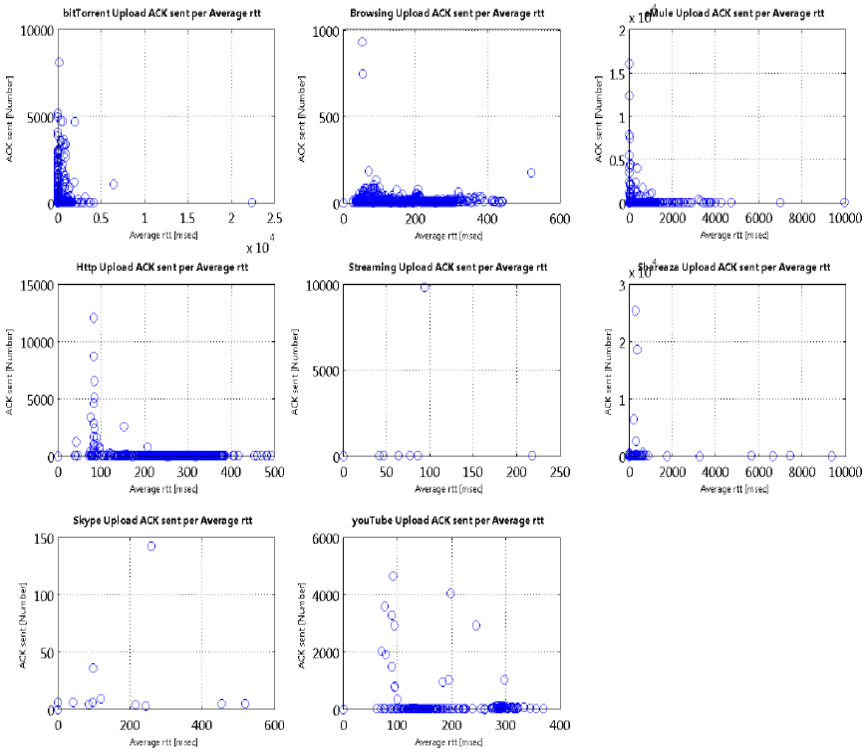

figura 24: *ACK Messages Upload per Average Round Trip Time*

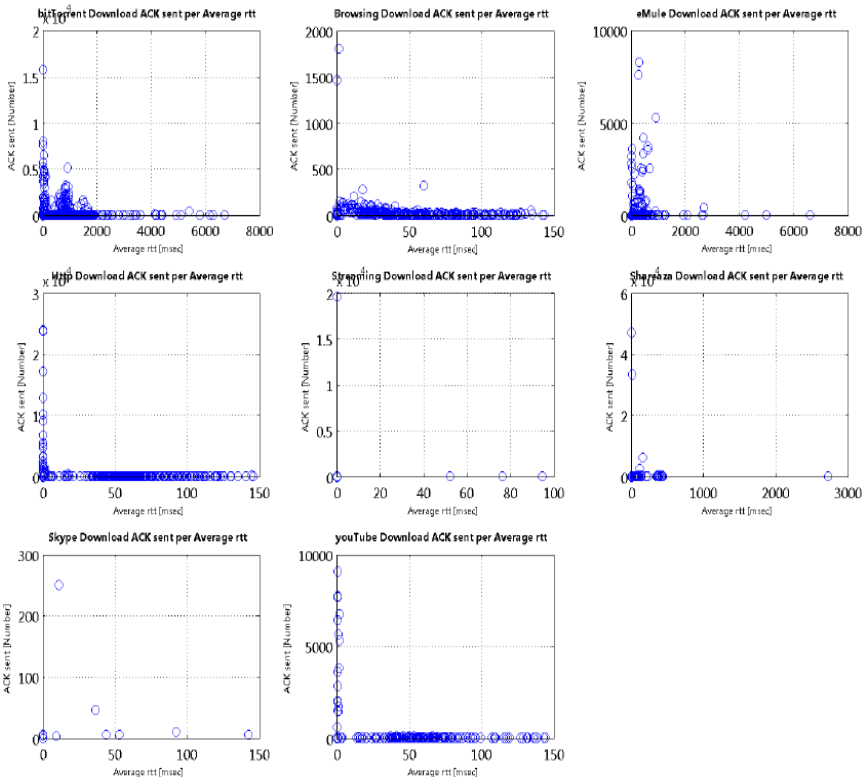

figura 25: *ACK Messages Download per Average Round Trip Time*

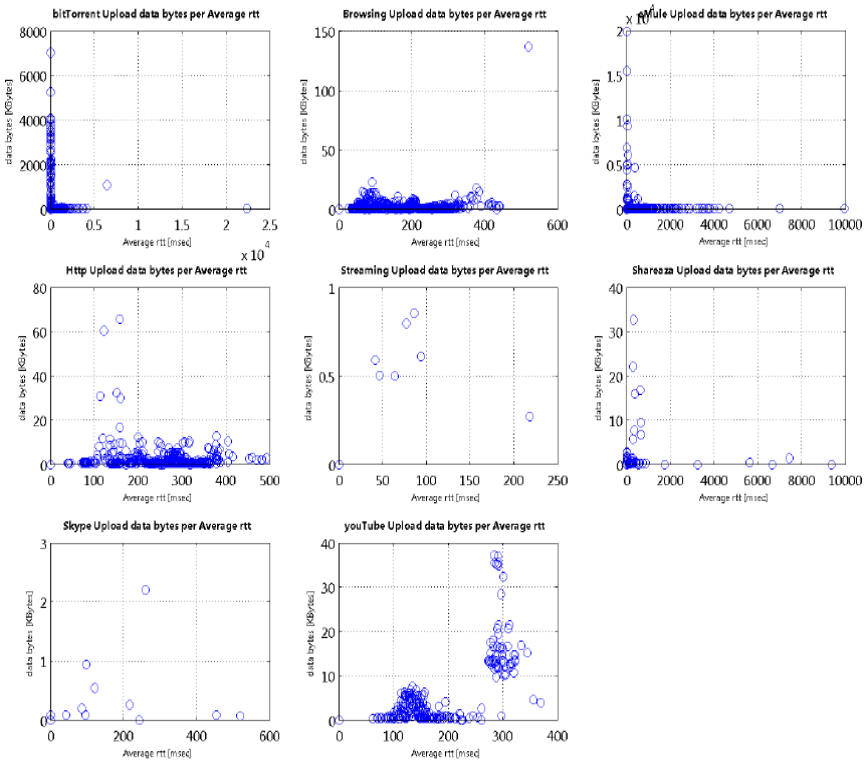

figura 26: *Data Bytes Upload per Average Round Trip Time*

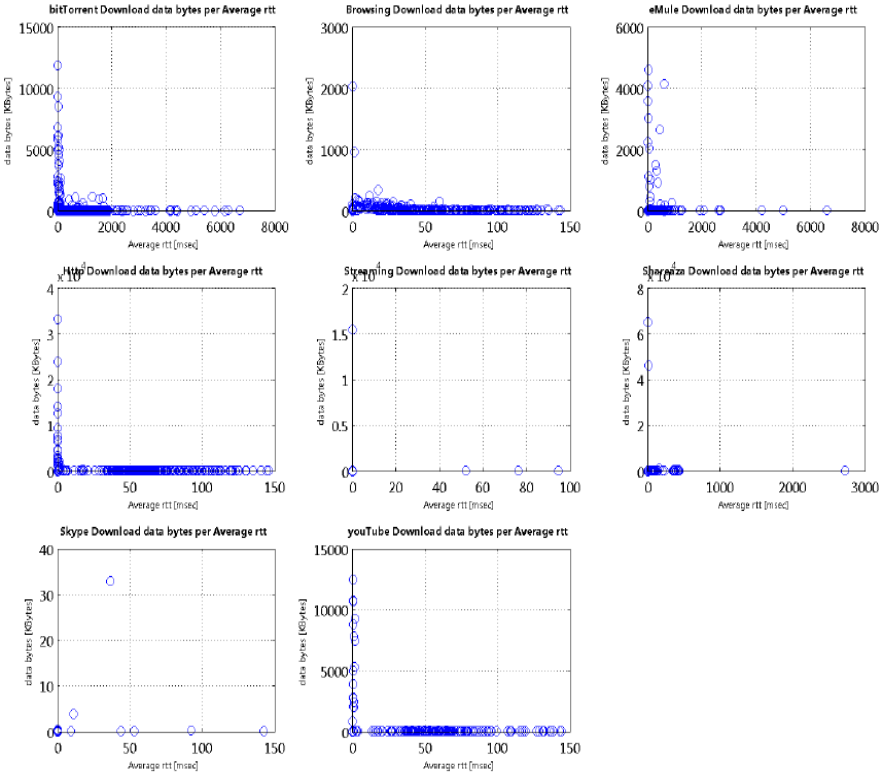

figura 27: *Data Bytes Download per Average Round Trip Time*

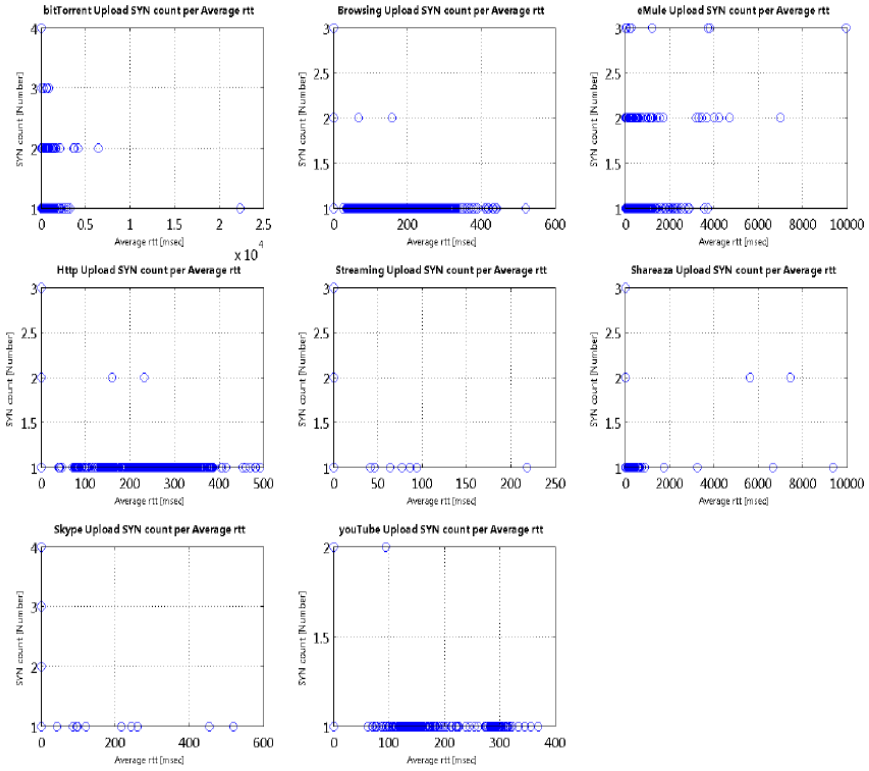

figura 28: *SYN Messages Upload per Average Round Trip Time*

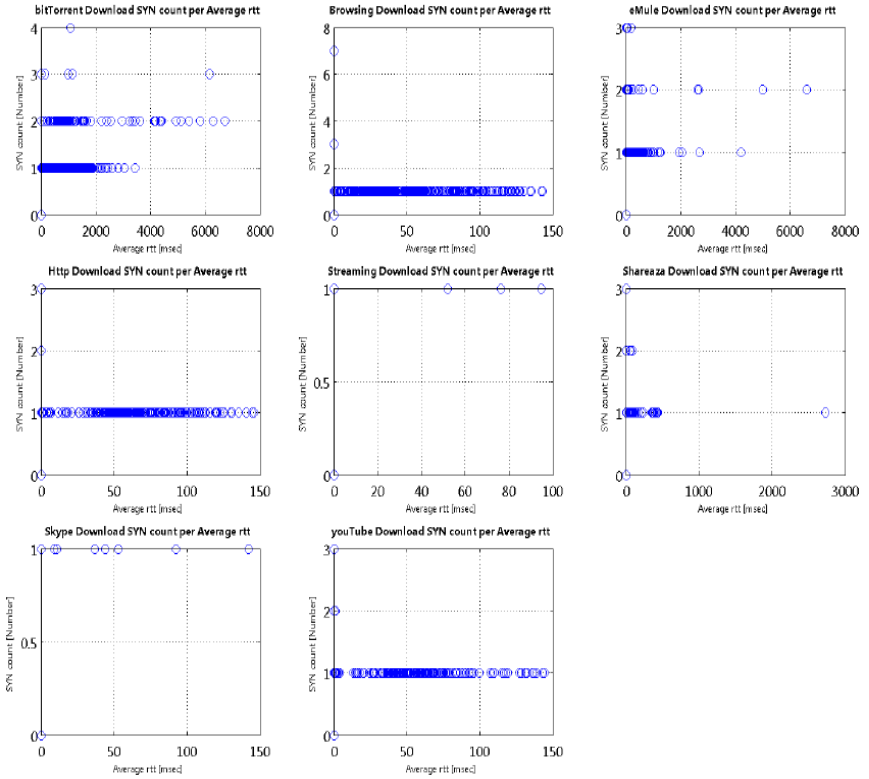

figura 29: *SYN Messages Download per Average Round Trip Time*

MENSAGENS DE SINCRONIZAÇÃO

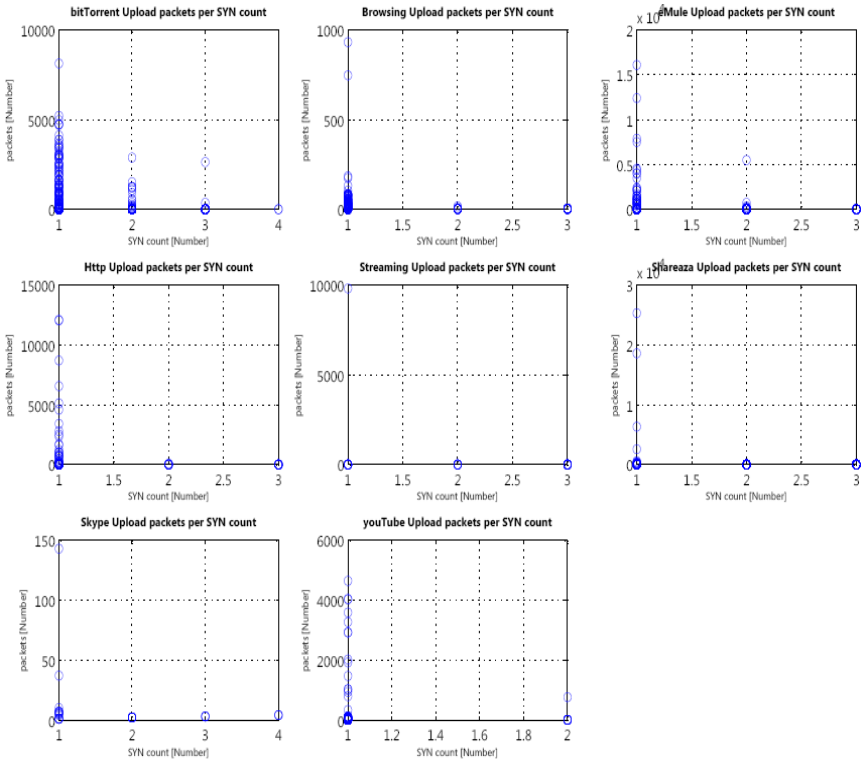

figura 30: *Packets Upload per SYN Messages*

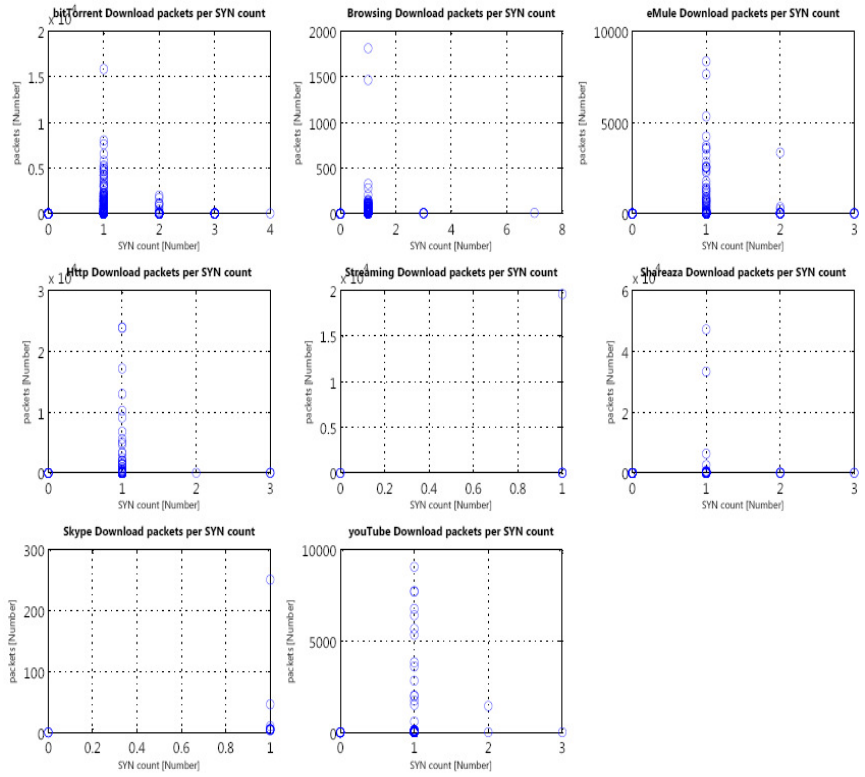

figura 31: *Packets Download per SYN Messages*

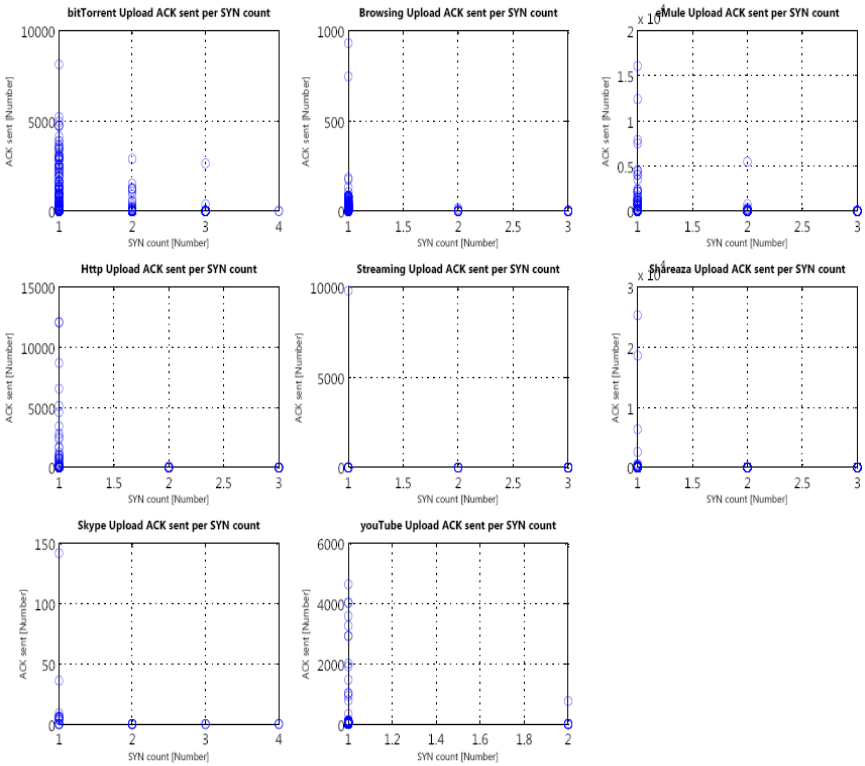

figura 32: *ACK Messages Upload per SYN Messages*

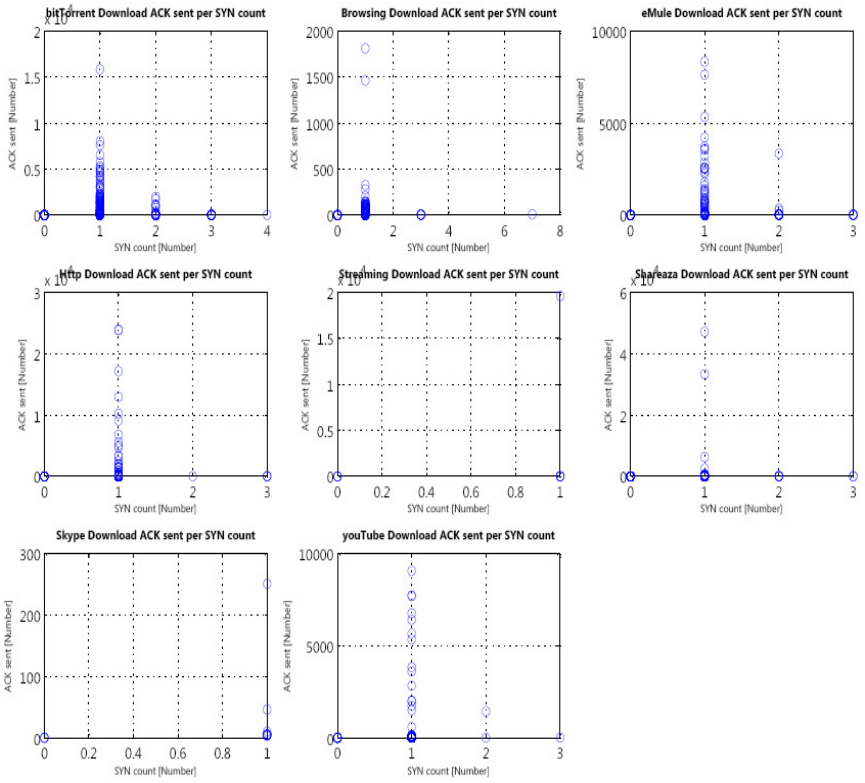

figura 33: *ACK Messages Download per SYN Messages*

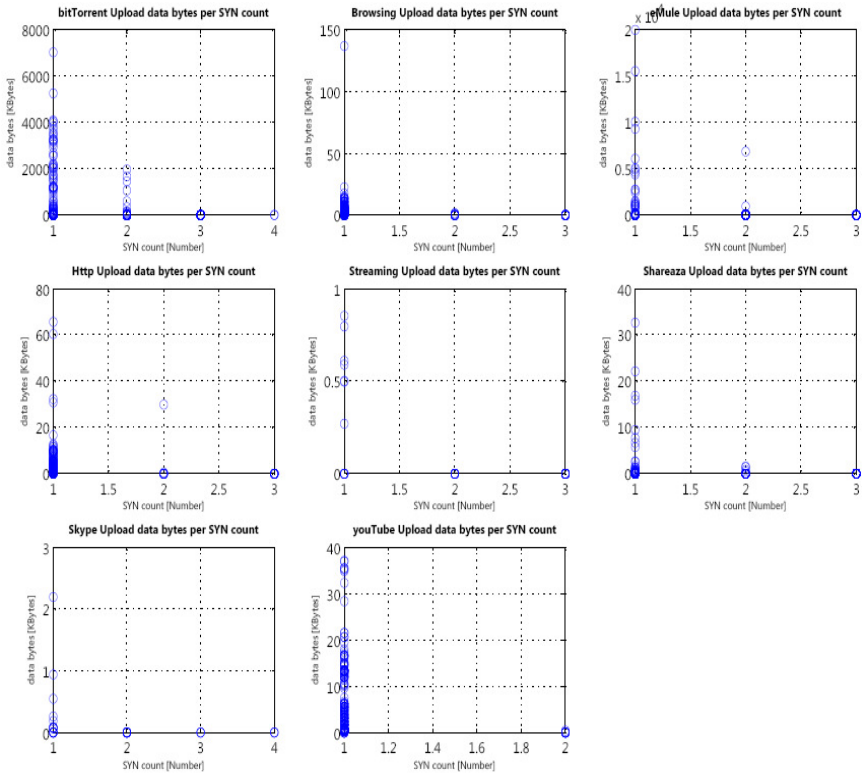

figura 34: *Data Bytes Upload per SYN Messages*

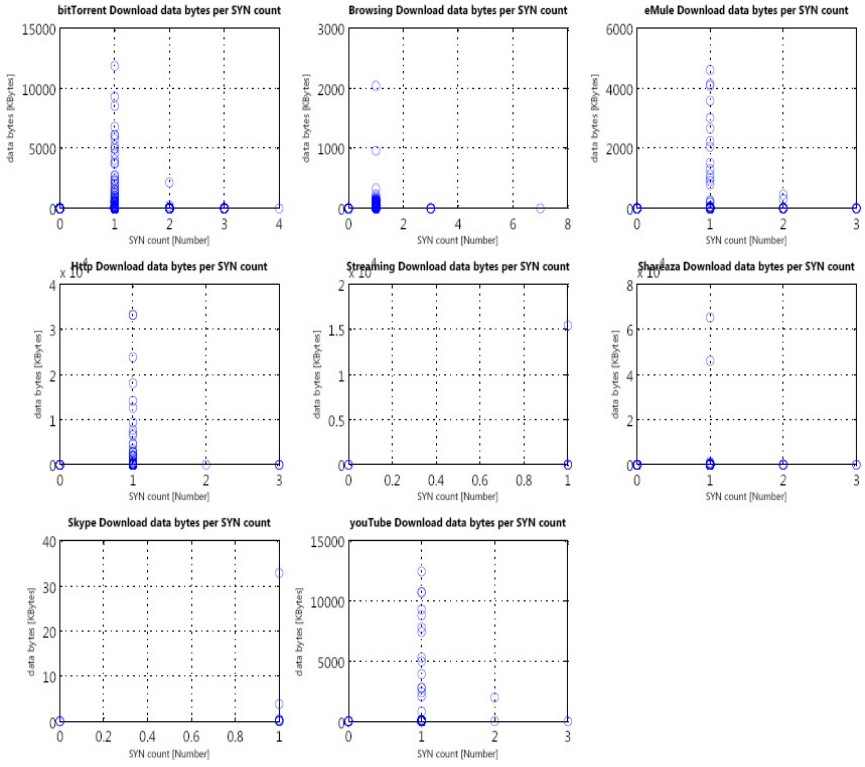

figura 35: *Data Bytes Download per SYN Messages*

VOLUME DE INFORMAÇÃO (*KBYTES*)

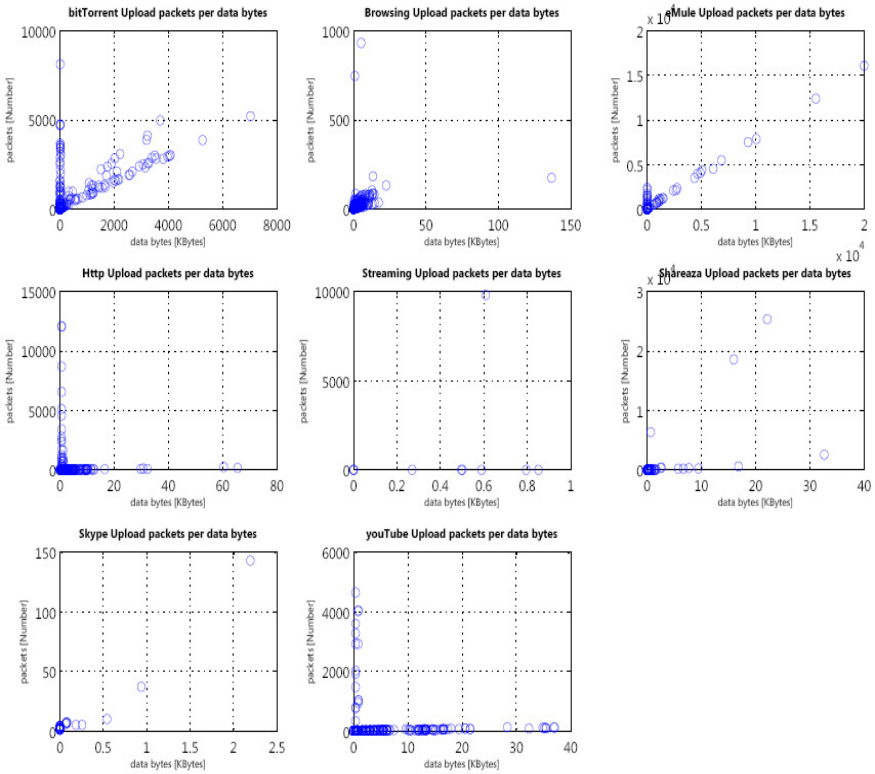

figura 36: ***Packets Upload per Data Bytes***

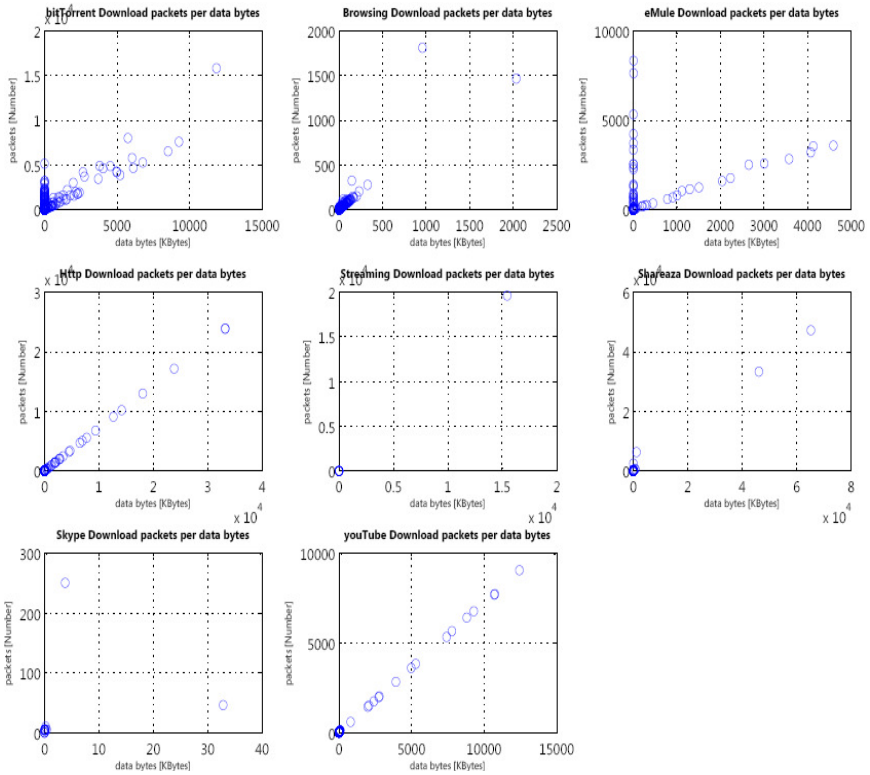

figura 37: ***Packets Download per Data Bytes***

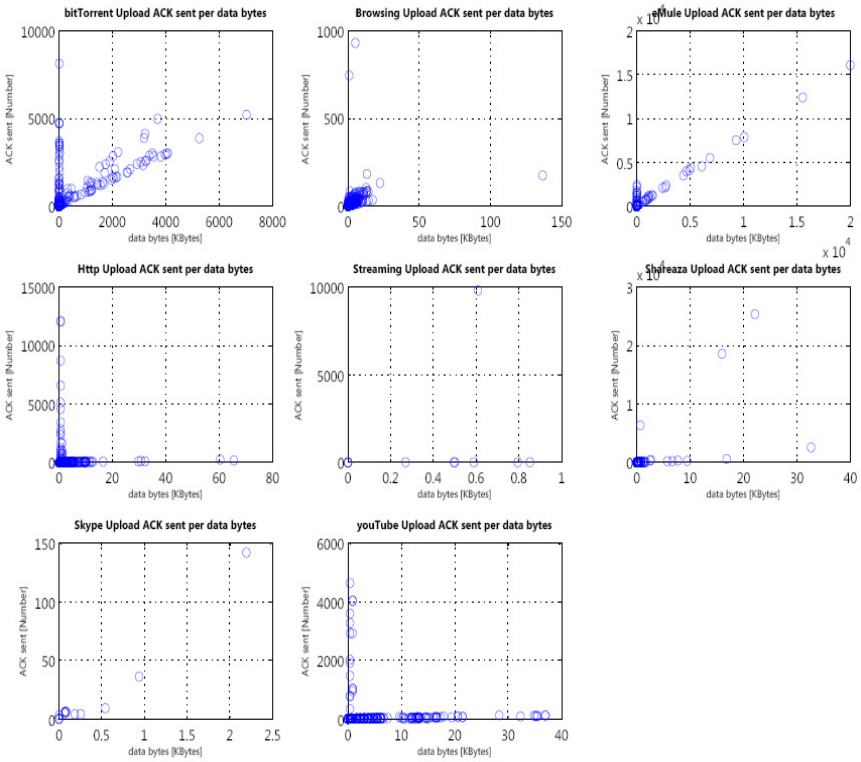

figura 38: *ACK Messages Upload per Data Bytes*

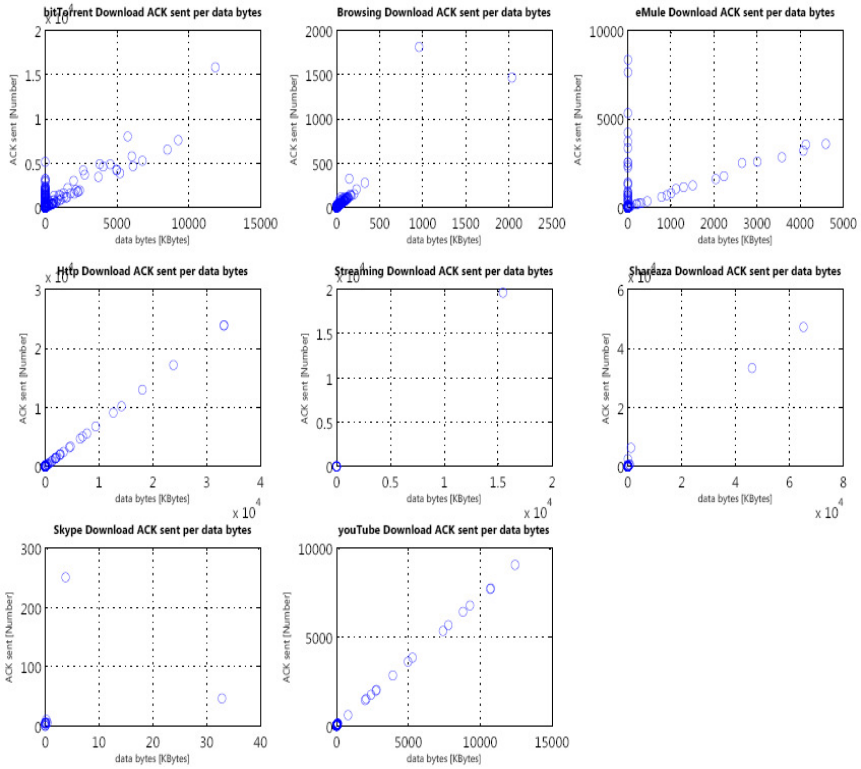

figura 39: *ACK Messages Download per Data Bytes*

MENSAGENS DE *ACKNOWLEDGE*

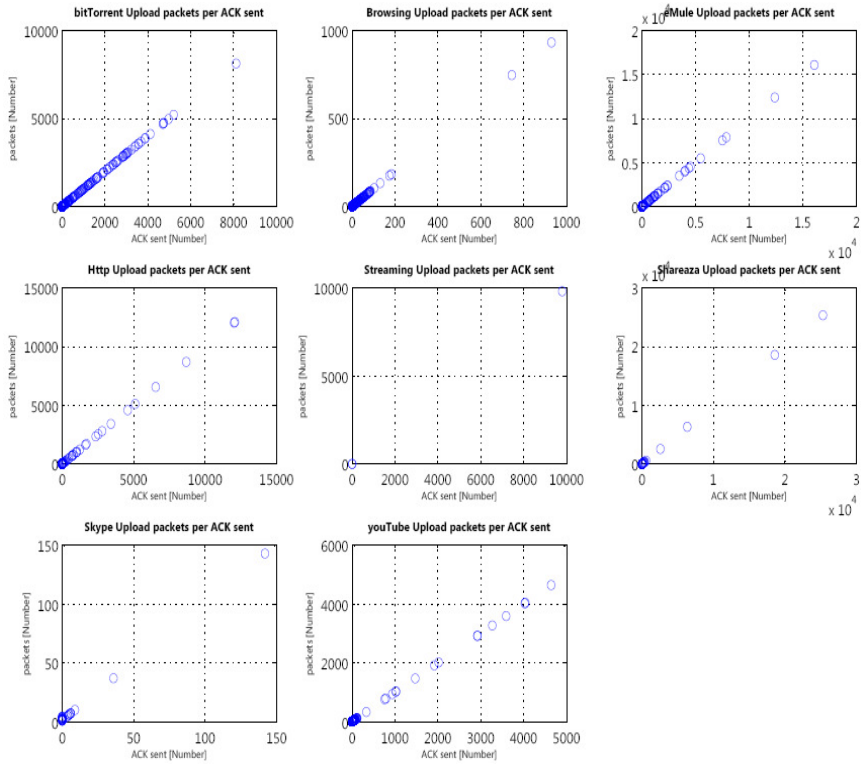

figura 40: *Packets Upload per ACK Messages*

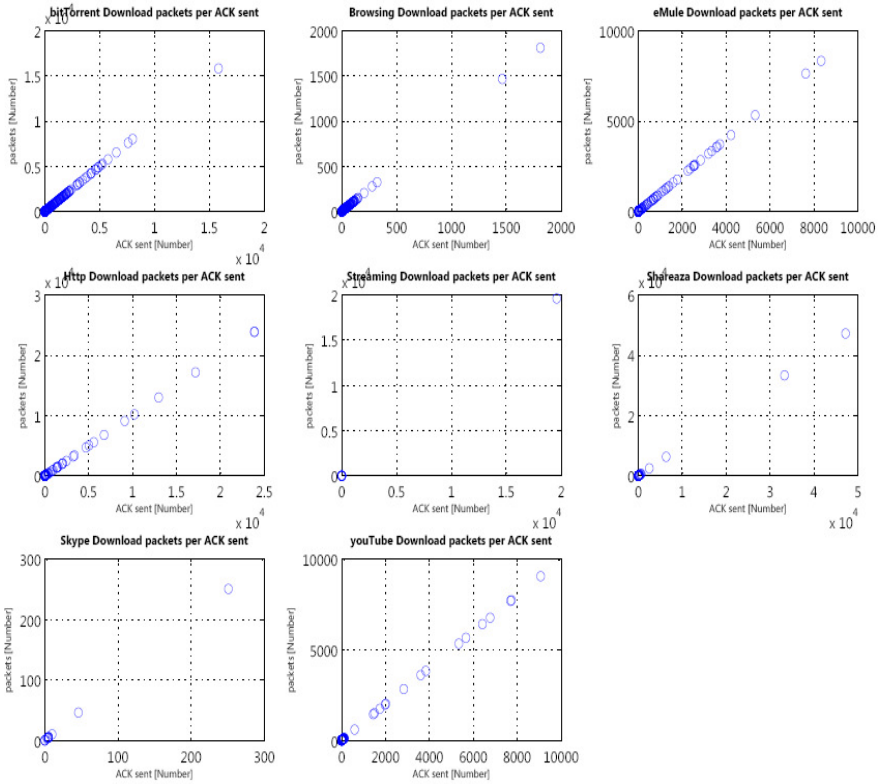

figura 41: *Packets Download per ACK Messages*

ANEXO B

Neste anexo é apresentado o código das *m-files* desenvolvidas no âmbito da dissertação.

BESTDATACOMBINATION.M

A identificação da combinação válida com maior número de parâmetros é feita pela seguinte função MATLAB:

```
function pca_data = bestDataCombination (initial_data_sets, component, valid_pct, report_params)

if report_params.write == 1
    mode = 'wt';
    fid = fopen(strcat(report_params.path, report_params.filename), mode);
end;


l_components_nbr = size(initial_data_sets(1).values);
l_data_sets_nbr = size(initial_data_sets);
l_comb_matrix = allPossibleCombinations (l_components_nbr(1,2));
l_comb_matrix_lines = size(l_comb_matrix);
l_valid_combination = 1;
for combination_line = 1:l_comb_matrix_lines(1,1)
    pca_data (l_valid_combination).components_nbr = 0;
    for data_component = 1:l_components_nbr(1,2)
        if l_comb_matrix(combination_line, data_component) ~= 0
            pca_data (l_valid_combination).components_nbr = pca_data
(l_valid_combination).components_nbr + 1;
        end;
    end;

    if pca_data (l_valid_combination).components_nbr > 2
        if report_params.write == 1
            fprintf(fid,
'################################################################################\n');
        end;
        l_first_component = 1;
        for data_component = 1:l_components_nbr(1,2)
            if l_comb_matrix(combination_line, data_component) ~= 0
                if l_first_component == 1
                    if report_params.write == 1
                        fprintf(fid, '# Combination line: %4d * Components: %s', l_valid_combination,
component(l_comb_matrix(combination_line, data_component)).name);
                    end;
                    l_first_component = 0;
                else
                    if report_params.write == 1
                        fprintf(fid, ', %s', component(l_comb_matrix(combination_line,
data_component)).name);
                    end;
                end;
            end;
        end;

        if report_params.write == 1
            fprintf(fid, '\n');
            fprintf(fid,
'#*******************************************************************************#\n');
        end;

        pca_data (l_valid_combination).valid_component_combination = 1;
        pca_data (l_valid_combination).less_pct = 100;
        for data_set_id = 1:l_data_sets_nbr(1,2)
            clear l_pca_data_set l_pca_component l_stdr l_data_lines l_sr l_pc l_zscores l_pcvars
l_cumulative;
            l_pca_component = 1;
            for data_component = 1:l_components_nbr(1,2)
                if l_comb_matrix(combination_line, data_component) ~= 0
```

```
                    l_pca_data_set.values (:,l_pca_component) =
initial_data_sets(data_set_id).values(:,l_comb_matrix(combination_line, data_component));
                    l_pca_component = l_pca_component + 1;
                end;
            end;
            l_stdr = std(l_pca_data_set.values);
            l_data_lines = size(l_pca_data_set.values);
            l_sr = l_pca_data_set.values./repmat(l_stdr, l_data_lines(1,1), 1);
            [l_pc, l_zscores, l_pcvars] = princomp(l_sr);
            l_cumulative = cumsum(l_pcvars./sum(l_pcvars) * 100);
            pca_data (l_valid_combination).set(data_set_id).values = l_zscores;
            pca_data (l_valid_combination).set(data_set_id).pct = l_cumulative(2,1);
            pca_data (l_valid_combination).set(data_set_id).pc_vectors = l_pc;
            pca_data (l_valid_combination).set(data_set_id).std_data = l_sr;
            if pca_data (l_valid_combination).set(data_set_id).pct < valid_pct
               pca_data (l_valid_combination).valid_component_combination = 0;
            end;
            if pca_data (l_valid_combination).less_pct > pca_data
(l_valid_combination).set(data_set_id).pct
                pca_data (l_valid_combination).less_pct = pca_data
(l_valid_combination).set(data_set_id).pct;
            end;
            if report_params.write == 1
               fprintf(fid, '# %s: %4.4f%%\n', initial_data_sets(data_set_id).name,
l_cumulative(2,1));
            end;
        end;
        if report_params.write == 1
            if pca_data (l_valid_combination).valid_component_combination == 0
               fprintf(fid, '# INVALID COMPONENT COMBINATION!\n');
            end;
            fprintf(fid,
'################################################################################\n');
        end;
        l_valid_combination = l_valid_combination + 1;
    end;
end;

if report_params.write == 1
    fprintf(fid, '                COMBINATION  | COMPONENTS NBR |    MIN PCT\n');
end;
for valid_comb_idx = 1:(l_valid_combination-1)
    if report_params.write == 1
        fprintf(fid, '                    %4d      |     %4d       |    %4.4f%%\n', valid_comb_idx,
pca_data (valid_comb_idx).components_nbr, pca_data (valid_comb_idx).less_pct);
    end;
end;

fclose all;
```

Recorre à função ***allPossibleCombinations*** que devolve uma matriz com as combinações possíveis para o número de parâmetros diferentes que compõem os dados estatísticos.

```
function combinations_set = allPossibleCombinations (max_nbr)

max_value = 0;
coefficient = max_nbr;
i = 0;
while i <= (max_nbr-1)
    max_value = max_value + coefficient*(10^i);
    coefficient = coefficient - 1;
    i = i + 1;
end;

line_nbr = 1;
i = 1;
while i <= max_value
    current_value = i;
    valid_value = 1;
    j = 0;
    while current_value ~= 0 && valid_value == 1
        j = j + 1;
        idx(j) = mod(current_value, 10);
        current_value = (current_value - idx(j))/10;

        if ((j > 1 && idx(j-1) <= idx(j)) || idx(j) > max_nbr)
            valid_value = 0;
        end;
    end;
    if valid_value == 1
        k = 0;
        while k < max_nbr
```

```
            k = k + 1;
            if k <= j
                combinations_set(line_nbr,k) = idx(k);
            else
                combinations_set(line_nbr,k) = 0;
            end;
        end;
        line_nbr = line_nbr + 1;
    end;
    i = i + 1;
end;
```

GETINCEDENCEAREAS.M

Esta função calcula as áreas relevantes de uma aplicação, isto é, as áreas onde o número de pontos das primeiras duas componentes principais do tráfego da aplicação é muito superior ao das restantes aplicações. Este processamento permite verificar se é possível ou não identificar o tráfego de uma qualquer aplicação no tráfego Internet.

```
function [data_subsets, total_areas] = getIncidenceAreas (data_sets, analyzing_set_id,
nonsignificant_pct, relevant_pct, squares_nbr)

data_sets_size = size(data_sets);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Inicializacao da Matriz final
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
total_areas = 1;
global_finished = 1;

values_size = size(data_sets(analyzing_set_id).values);
data_subsets(1).finished = 0;
data_subsets(1).relevant_area = 1;
data_subsets(1).minvalues(1,1) = min(data_sets(analyzing_set_id).values(:,1));
data_subsets(1).minvalues(1,2) = min(data_sets(analyzing_set_id).values(:,2));
data_subsets(1).maxvalues(1,1) = max(data_sets(analyzing_set_id).values(:,1));
data_subsets(1).maxvalues(1,2) = max(data_sets(analyzing_set_id).values(:,2));

% Analyzing set
data_subsets(1).set(analyzing_set_id).values = data_sets(analyzing_set_id).values;
data_subsets(1).set(analyzing_set_id).occurrences = values_size(1,1);
data_subsets(1).set(analyzing_set_id).total_points = values_size(1,1);
if data_subsets(1).set(analyzing_set_id).occurrences <= 0
   data_subsets(1).finished = 1;
   data_subsets(1).relevant_area = 0;
end;

% Other sets
for set_id = 1:data_sets_size(1,2)
    if set_id ~= analyzing_set_id
        clear values_size;
        values_size = size(data_sets(set_id).values);
        data_subsets(1).minvalues(1,1) = min([data_subsets(1).minvalues(1,1)
min(data_sets(set_id).values(:,1))]);
        data_subsets(1).minvalues(1,2) = min([data_subsets(1).minvalues(1,2)
min(data_sets(set_id).values(:,2))]);
        data_subsets(1).maxvalues(1,1) = max([data_subsets(1).maxvalues(1,1)
max(data_sets(set_id).values(:,1))]);
        data_subsets(1).maxvalues(1,2) = max([data_subsets(1).maxvalues(1,2)
max(data_sets(set_id).values(:,2))]);
        data_subsets(1).set(set_id).values = data_sets(set_id).values;
        data_subsets(1).set(set_id).occurrences = values_size(1,1);
        data_subsets(1).set(set_id).total_points = values_size(1,1);
        if data_subsets(1).set(set_id).occurrences > 0 && data_subsets(1).finished ~= 1
            if ((data_subsets(1).set(set_id).occurrences/data_subsets(1).set(set_id).total_points) >
(nonsignificant_pct/100)) &&
((data_subsets(1).set(analyzing_set_id).occurrences/data_subsets(1).set(analyzing_set_id).total_point
s) >= (relevant_pct/100))
                global_finished = 0;
                data_subsets(1).relevant_area = 0;
            else
                if
((data_subsets(1).set(set_id).occurrences/data_subsets(1).set(set_id).total_points) >
(nonsignificant_pct/100))
                    data_subsets(1).finished = 1;
                    data_subsets(1).relevant_area = 0;
                else
```

```
                    data_subsets(1).relevant_area = 1;
                end;
            end;
        end;
    end;
end;

while global_finished == 0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reset da estrutura usada no calculo
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    clear previous_total_areas previous_data_subsets;
    previous_total_areas = total_areas;
    previous_data_subsets = data_subsets;
    clear total_areas data_subsets;
    global_finished = 1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Re-calculo de valores para as novas areas
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    total_areas = 0;
    for areaid = 1:previous_total_areas
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Verificacao de areas finalizadas e transferencia para estrutura final
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        if previous_data_subsets(areaid).finished == 1
           total_areas = total_areas + 1;
           data_subsets (total_areas) = previous_data_subsets(areaid);
        else
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determinacao do tamanho da nova area
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            x_hope = (previous_data_subsets(areaid).maxvalues(1,1) -
previous_data_subsets(areaid).minvalues(1,1))/squares_nbr;
            y_hope = (previous_data_subsets(areaid).maxvalues(1,2) -
previous_data_subsets(areaid).minvalues(1,2))/squares_nbr;
            matrix_x_cursor = previous_data_subsets(areaid).minvalues(1,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Percorrer cada area segundo x (esquerda para a direita)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            while matrix_x_cursor <= previous_data_subsets(areaid).maxvalues(1,1)
                matrix_y_cursor = previous_data_subsets(areaid).minvalues(1,2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Percorrer cada area segundo y (baixo para cima)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                while matrix_y_cursor <= previous_data_subsets(areaid).maxvalues(1,2)
                    total_areas = total_areas + 1;
                    data_subsets(total_areas).minvalues = [matrix_x_cursor matrix_y_cursor];
                    data_subsets(total_areas).maxvalues = [matrix_x_cursor+x_hope
matrix_y_cursor+y_hope];
                    data_subsets(total_areas).finished = 0;
                    data_subsets(total_areas).relevant_area = 1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determinacao dos pontos pertencentes a nova area
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Analyzing set
                    data_subsets(total_areas).set(analyzing_set_id).total_points =
previous_data_subsets(areaid).set(analyzing_set_id).total_points;
                    data_subsets(total_areas).set(analyzing_set_id).occurrences = 0;
                    for values_line =
1:previous_data_subsets(areaid).set(analyzing_set_id).occurrences
                        if previous_data_subsets(areaid).set(analyzing_set_id).values(values_line,1)
>= data_subsets(total_areas).minvalues(1,1) &&
previous_data_subsets(areaid).set(analyzing_set_id).values(values_line,1) <
data_subsets(total_areas).maxvalues(1,1) &&
previous_data_subsets(areaid).set(analyzing_set_id).values(values_line,2) >=
data_subsets(total_areas).minvalues(1,2) &&
previous_data_subsets(areaid).set(analyzing_set_id).values(values_line,2) <
data_subsets(total_areas).maxvalues(1,2)
                            data_subsets(total_areas).set(analyzing_set_id).occurrences =
data_subsets(total_areas).set(analyzing_set_id).occurrences + 1;

data_subsets(total_areas).set(analyzing_set_id).values(data_subsets(total_areas).set(analyzing_set_id
).occurrences,:) = previous_data_subsets(areaid).set(analyzing_set_id).values(values_line,:);
                        end;
                    end;
                    if data_subsets(total_areas).set(analyzing_set_id).occurrences <= 0
                       data_subsets(total_areas).finished = 1;
                       data_subsets(total_areas).relevant_area = 0;
                    elseif
((data_subsets(total_areas).set(analyzing_set_id).occurrences/data_subsets(total_areas).set(analyzing
_set_id).total_points) < (relevant_pct/100))
                       data_subsets(total_areas).finished = 1;
                       data_subsets(total_areas).relevant_area = 0;
                    end;
```

```
% Other sets
                    for set_id = 1:data_sets_size(1,2)
                        if set_id ~= analyzing_set_id
                            data_subsets(total_areas).set(set_id).total_points =
previous_data_subsets(areaid).set(set_id).total_points;
                            data_subsets(total_areas).set(set_id).occurrences = 0;
                            for values_line = 1:previous_data_subsets(areaid).set(set_id).occurrences
                                if previous_data_subsets(areaid).set(set_id).values(values_line,1) >=
data_subsets(total_areas).minvalues(1,1) &&
previous_data_subsets(areaid).set(set_id).values(values_line,1) <
data_subsets(total_areas).maxvalues(1,1) &&
previous_data_subsets(areaid).set(set_id).values(values_line,2) >=
data_subsets(total_areas).minvalues(1,2) &&
previous_data_subsets(areaid).set(set_id).values(values_line,2) <
data_subsets(total_areas).maxvalues(1,2)
                                    data_subsets(total_areas).set(set_id).occurrences =
data_subsets(total_areas).set(set_id).occurrences + 1;

data_subsets(total_areas).set(set_id).values(data_subsets(total_areas).set(set_id).occurrences,:) =
previous_data_subsets(areaid).set(set_id).values(values_line,:);
                                end;
                            end;
                            if data_subsets(total_areas).set(set_id).occurrences > 0 &&
data_subsets(total_areas).finished ~= 1
                                if
((data_subsets(total_areas).set(set_id).occurrences/data_subsets(total_areas).set(set_id).total_point
s) > (nonsignificant_pct/100)) &&
((data_subsets(total_areas).set(analyzing_set_id).occurrences/data_subsets(total_areas).set(analyzing
_set_id).total_points) >= (relevant_pct/100))
                                    global_finished = 0;
                                    data_subsets(1).relevant_area = 0;
                                else
                                    if
((data_subsets(total_areas).set(set_id).occurrences/data_subsets(total_areas).set(set_id).total_point
s) > (nonsignificant_pct/100))
                                        data_subsets(total_areas).finished = 1;
                                        data_subsets(total_areas).relevant_area = 0;
                                    else
                                        data_subsets(total_areas).relevant_area = 1;
                                    end;
                                end;
                            end;
                        end;
                    end;
                    matrix_y_cursor = matrix_y_cursor + y_hope;
                end;
                matrix_x_cursor = matrix_x_cursor + x_hope;
            end;
        end;
    end;
end;
```

INCEDENCEAREASREPORT.M

Função responsável pela geração de um relatório relativo às áreas de predominância de cada aplicação presente no conjunto de dados estatísticos.

```
function data_analysis = incidenceAreasReport (data_sets, nonsignificant_pct, relevant_pct,
squares_set, report_params)

currentdate = datestr(now, 'dd-mm-yyyy HH:MM:SS');
if report_params.write == 1
    mode = 'wt';
    fid = fopen(strcat(report_params.path, report_params.filename), mode);
end;

data_sets_size = size(data_sets);
squares_set_size = size(squares_set);
if data_sets_size(1,1) == 1 && data_sets_size(1,2) > 1 && nonsignificant_pct > 0 && relevant_pct > 0
&& squares_set_size(1,1) == 1 && squares_set_size(1,2) == 2 && squares_set(1,1) <= squares_set(1,2)
    if report_params.write == 1
        fprintf(fid,
'################################################################################\n');
        fprintf(fid, '# INCIDENCE AREAS REPORT at %s           Data Sets Nbr: %3d   #\n',
currentdate, data_sets_size(1,2));
        fprintf(fid,
'################################################################################\n');
        fprintf(fid, '* Non Significant Pct: %02.2f%%    *     Min Sqr Nbr  : %3d   \n',
nonsignificant_pct, squares_set(1,1));
```

```
        fprintf(fid, '* Relevant Pct       : %02.2f%%    *     Max Sqr Nbr  : %3d   \n',
relevant_pct, squares_set(1,2));
    end;

    for squares_nbr = squares_set(1,1):squares_set(1,2)
        for analyzing_set_id = 1:data_sets_size(1,2)
            clear data_subsets total_areas;
            [data_subsets, total_areas] = getIncidenceAreas (data_sets, analyzing_set_id,
nonsignificant_pct, relevant_pct, squares_nbr);
            data_analysis(analyzing_set_id).aproximation(squares_nbr).data_subsets = data_subsets;
            data_analysis(analyzing_set_id).aproximation(squares_nbr).total_areas = total_areas;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Geracao do relatorio, para o set id actual
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            if report_params.write == 1
               for set_id = 1:data_sets_size(1,2)
                    aggregated_values(set_id).occurrences = 0;
                    aggregated_values(set_id).total_points =
data_subsets(1).set(set_id).total_points;
               end;
               if analyzing_set_id == 1 && squares_nbr == squares_set(1,1)
                   fprintf(fid, '---------------------------\n');
                   fprintf(fid, '  Set ID  |  Total Points  \n');
                   fprintf(fid, '----------|---------------\n');
                   for set_id = 1:data_sets_size(1,2)
                       fprintf(fid, '   %4d   |    %7d     \n', set_id,
data_subsets(1).set(set_id).total_points);
                   end;
                   fprintf(fid, '---------------------------\n');
               end;
               fprintf(fid,
'\n#############################################################################\n');
               fprintf(fid, '# ANALYZING SET ID: %4d                                    SQUARES NBR:
%3d   \n', analyzing_set_id, squares_nbr);
               fprintf(fid,
'#############################################################################\n');


               for area_id = 1:total_areas
                   if data_subsets(area_id).relevant_area == 1
                       fprintf(fid, '****************\n');
                       fprintf(fid, ' AREA ID: %4d *  (min_x, min_y): (%6.4f, %6.4f)  (max_x,
max_y): (%6.4f, %6.4f)\n', area_id, data_subsets(area_id).minvalues(1,1),
data_subsets(area_id).minvalues(1,2), data_subsets(area_id).maxvalues(1,1),
data_subsets(area_id).maxvalues(1,2));
                       fprintf(fid,
'*****************************************************************************\n');
                       fprintf(fid, '                                                | SetID
|Occurrences|Contained Pct\n');
                       fprintf(fid, '-----------------------------------------------|-------|---------
--|-------------\n');
                       for set_id = 1:data_sets_size(1,2)
                           fprintf(fid, '                                                | %5d |  %7d
|     %03.2f%%   \n', set_id, data_subsets(area_id).set(set_id).occurrences,
(data_subsets(area_id).set(set_id).occurrences/data_subsets(area_id).set(set_id).total_points)*100);
                           aggregated_values(set_id).occurrences =
aggregated_values(set_id).occurrences + data_subsets(area_id).set(set_id).occurrences;
                       end;
                       fprintf(fid, '-----------------------------------------------|-------|---------
--|-------------\n');
                       fprintf(fid, '                              AGGREGATED VALUES|       |
|             \n');

                       for set_id = 1:data_sets_size(1,2)
                           fprintf(fid, '                                                | %5d |  %7d
|     %03.2f%%   \n', set_id, aggregated_values(set_id).occurrences,
(aggregated_values(set_id).occurrences/aggregated_values(set_id).total_points)*100);
                       end;
                       fprintf(fid, '-----------------------------------------------------------------
----------------\n');
                   end;
               end;
            end;
        end;
    end;
end;
fclose all;
```

ANEXO C

Os gráficos seguintes contêm a representação das duas primeiras componentes principais de cada aplicação, para os sub-conjuntos e conjunto original mencionados no capítulo 5. Refira-se que para a geração dos sub-conjuntos foi desenvolvida a seguinte função:

```
function data_subsets = createDataSubSets (initial_data_sets, subsets_nbr)
l_data_sets_nbr = size(initial_data_sets);
for set_id = 1:l_data_sets_nbr(1,2)
    l_set_values_size = size(initial_data_sets(set_id).values);
    l_subset_values_size = round(l_set_values_size(1,1)/subsets_nbr);
    l_subset_id = 1;
    l_subset_line = 1;
    for values_line_id = 1:l_set_values_size(1,1)
        if l_subset_line == 1
           data_subsets(l_subset_id).data(set_id).name = initial_data_sets(set_id).name;
        end;
        data_subsets(l_subset_id).data(set_id).values(l_subset_line,:) =
initial_data_sets(set_id).values(values_line_id,:);
        if mod(values_line_id, l_subset_values_size) == 0
            l_subset_id = l_subset_id + 1;
            l_subset_line = 1;
        else
           l_subset_line = l_subset_line + 1;
        end;
    end;
end;
```

BITTORRENT PCA

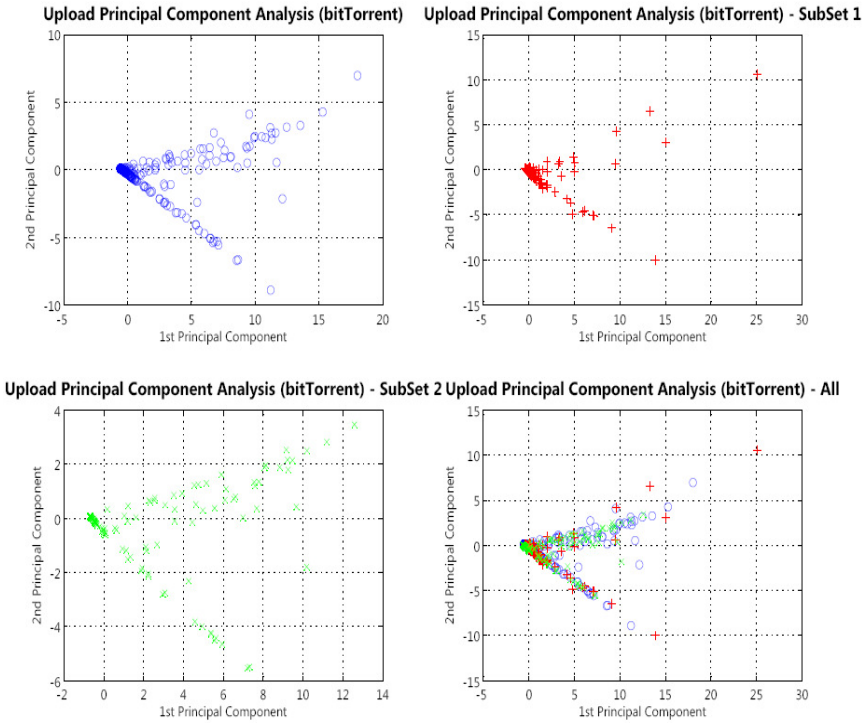

figura 42: *bitTorrent Upload PCA*

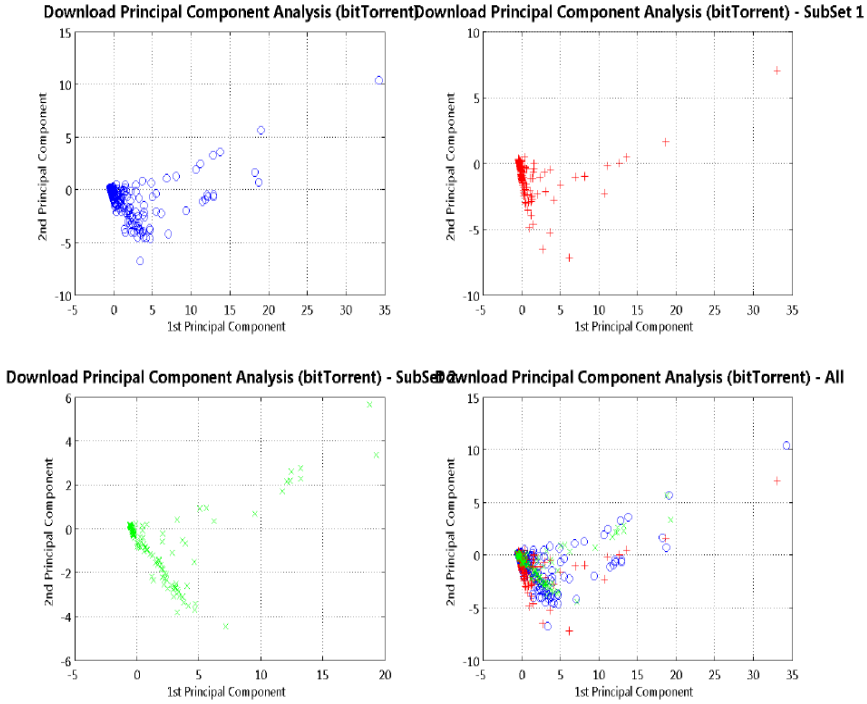

figura 43: *bitTorrent Download PCA*

BROWSING PCA

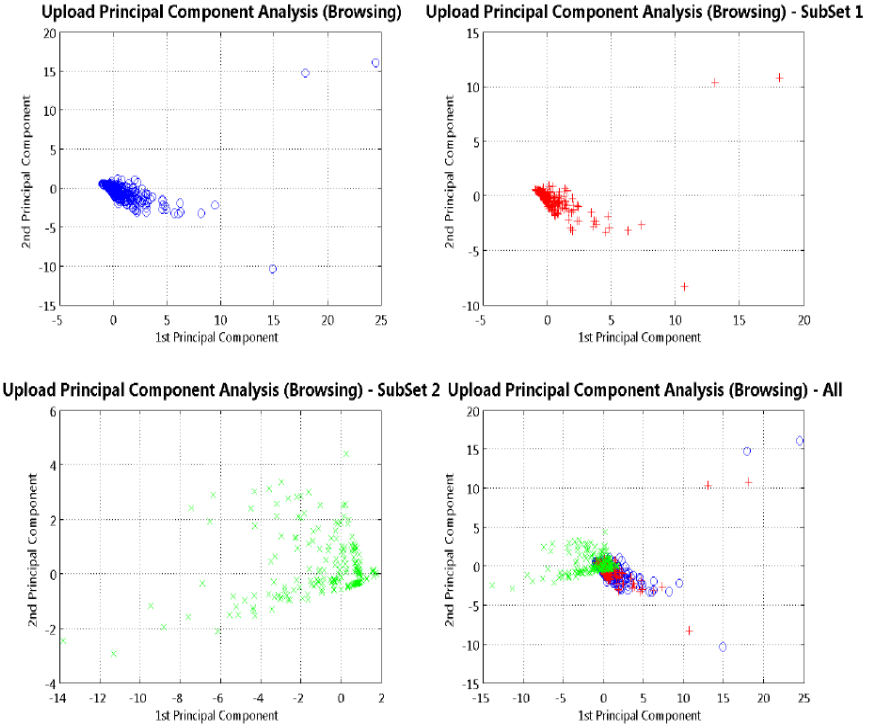

figura 44: *Browsing Upload PCA*

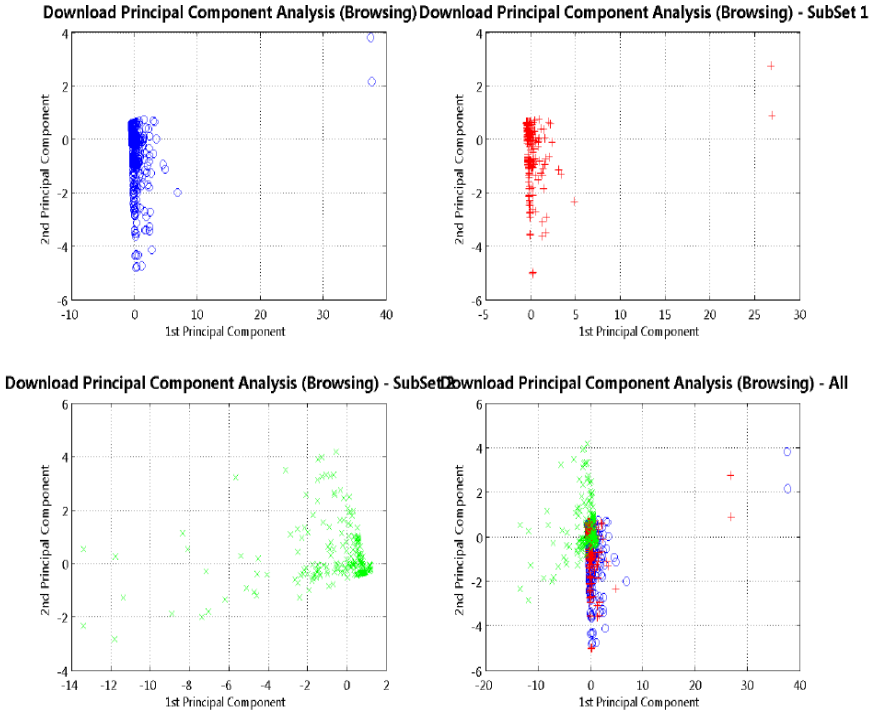

figura 45: *Browsing Download PCA*

EMULE PCA

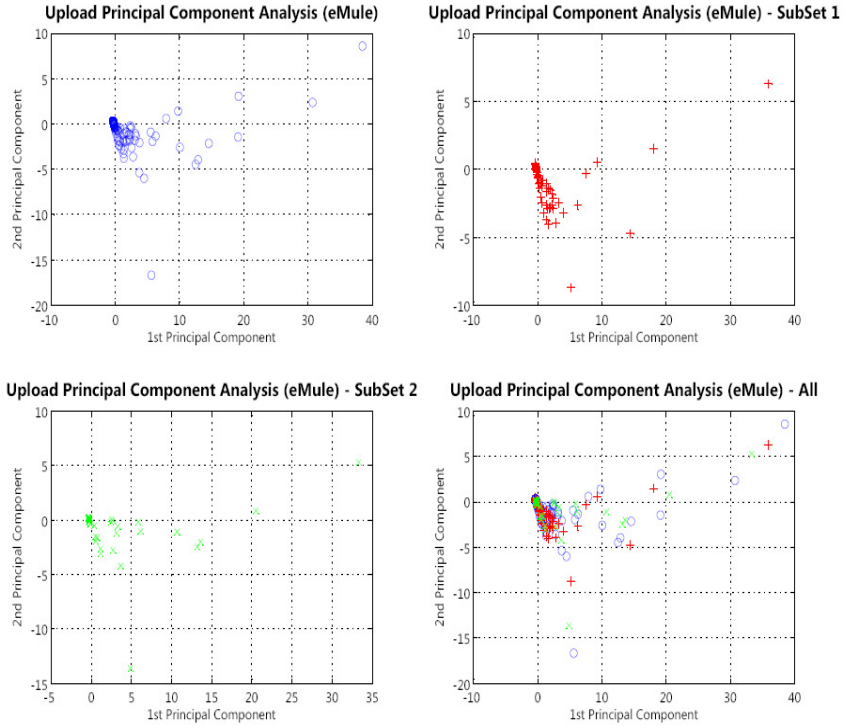

figura 46: *eMule Upload PCA*

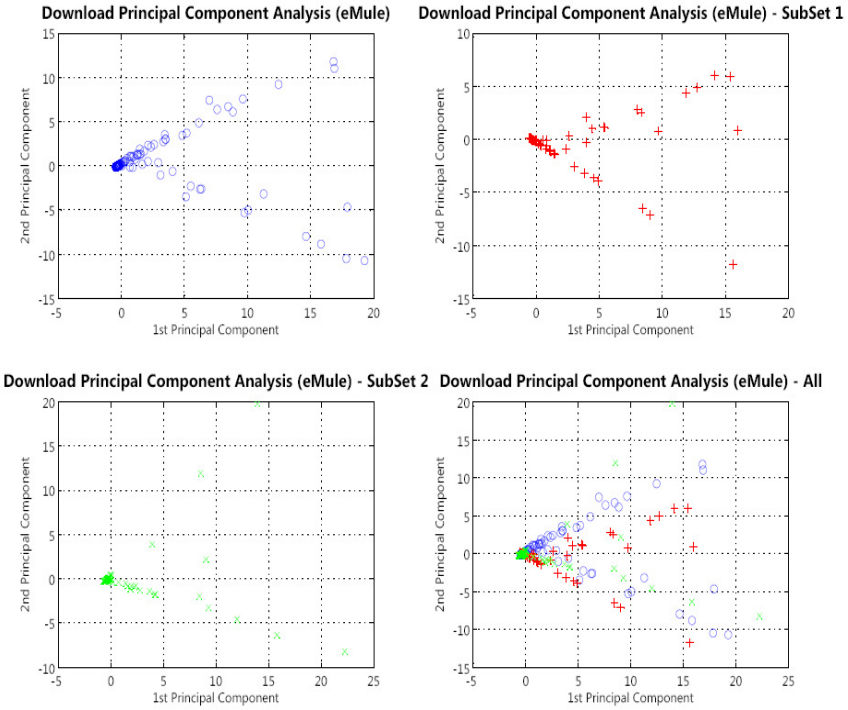

figura 47: *eMule Download PCA*

HTTP PCA

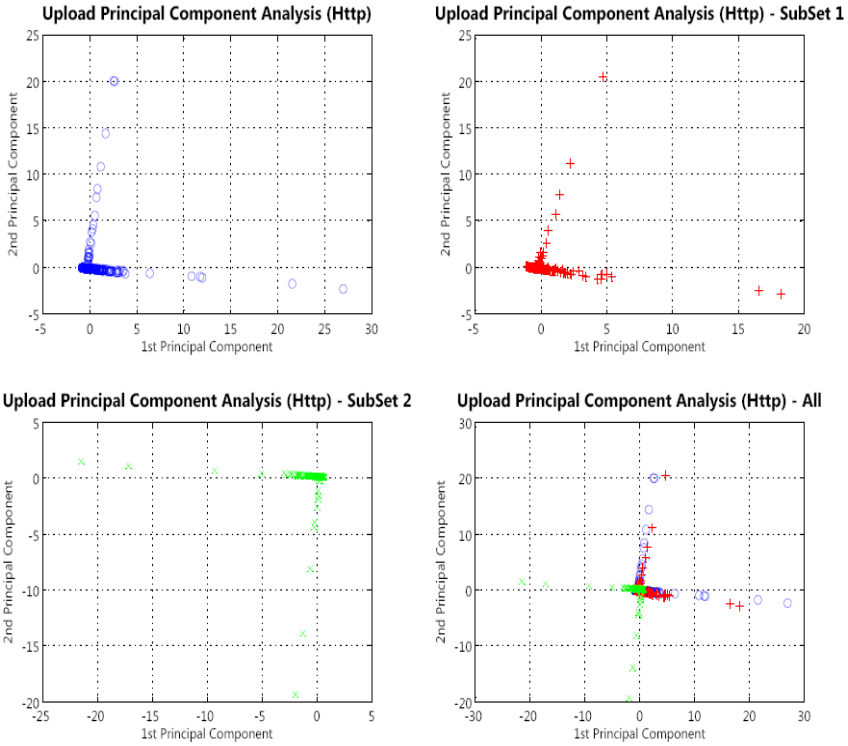

figura 48: *HTTP Upload PCA*

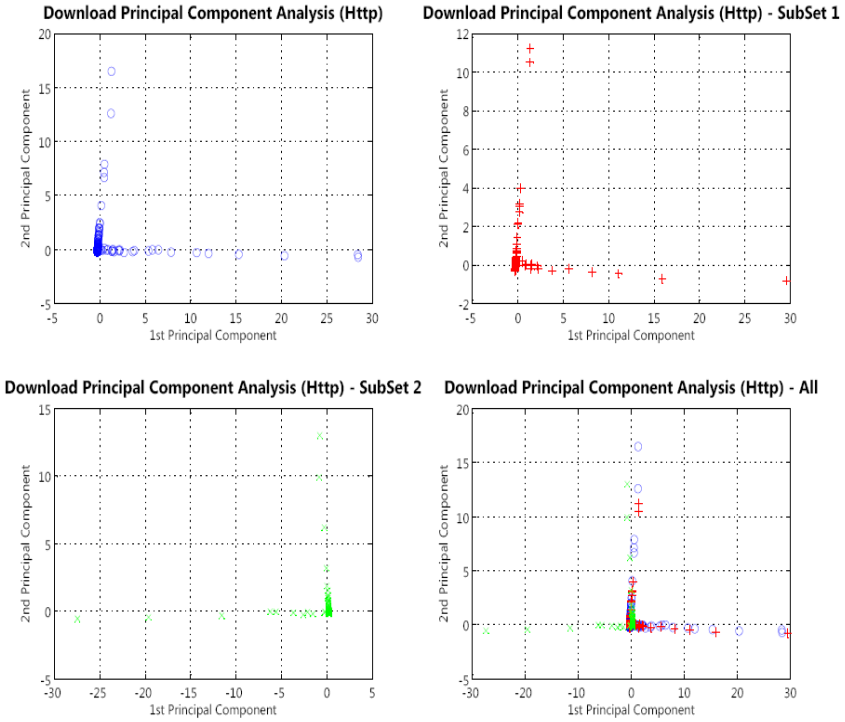

figura 49: *HTTP Download PCA*

STREAMING PCA

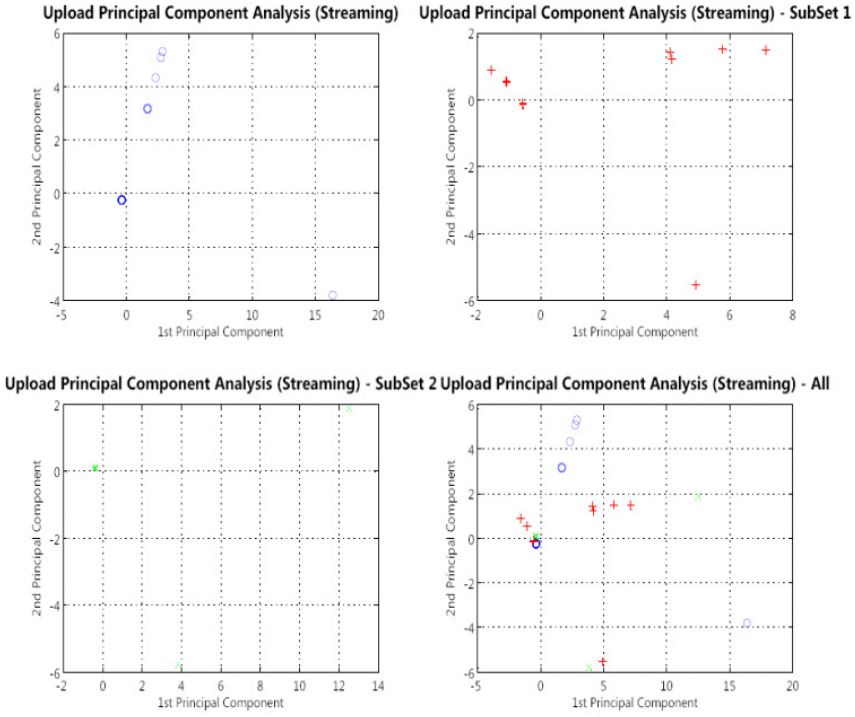

figura 50: *Streaming Upload PCA*

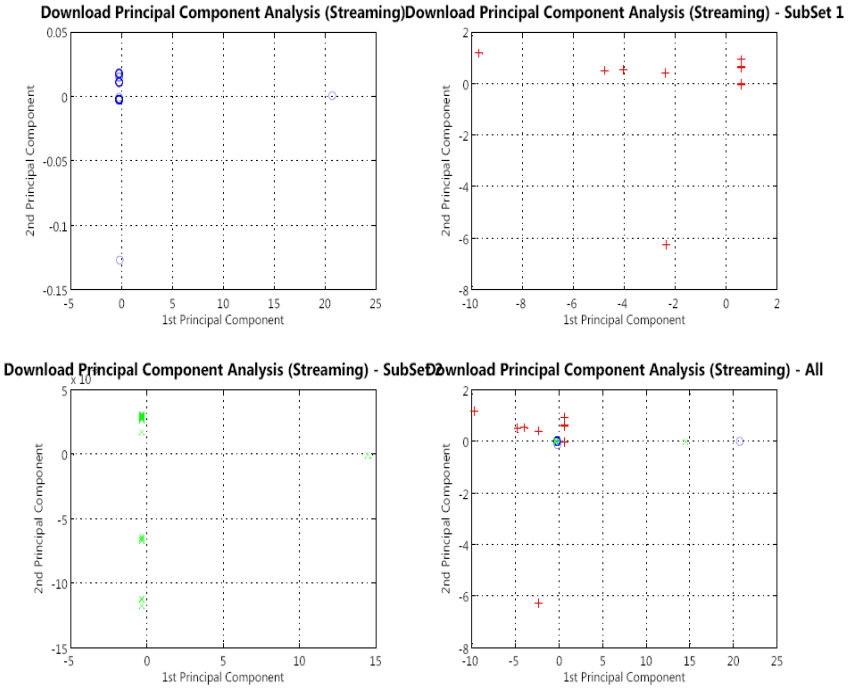

figura 51: *Streaming Download PCA*

SHAREAZA PCA

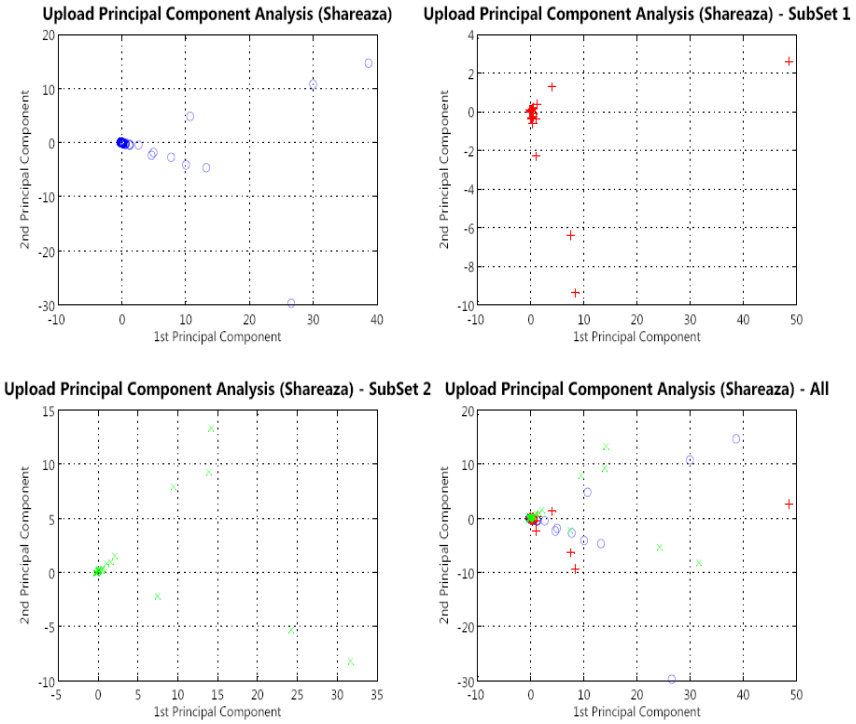

figura 52: *Shareaza Upload PCA*

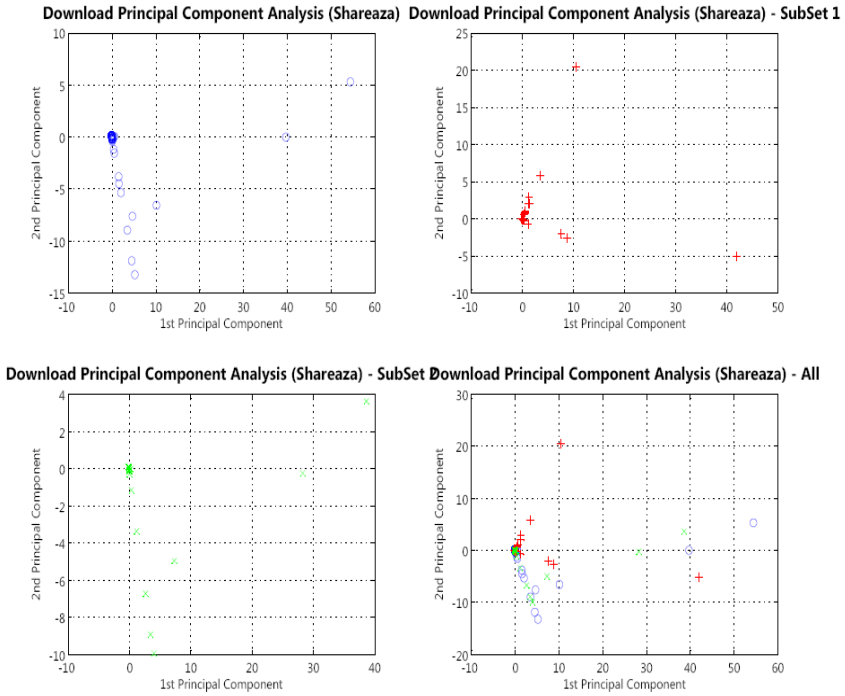

figura 53: *Shareaza Download PCA*

SKYPE PCA

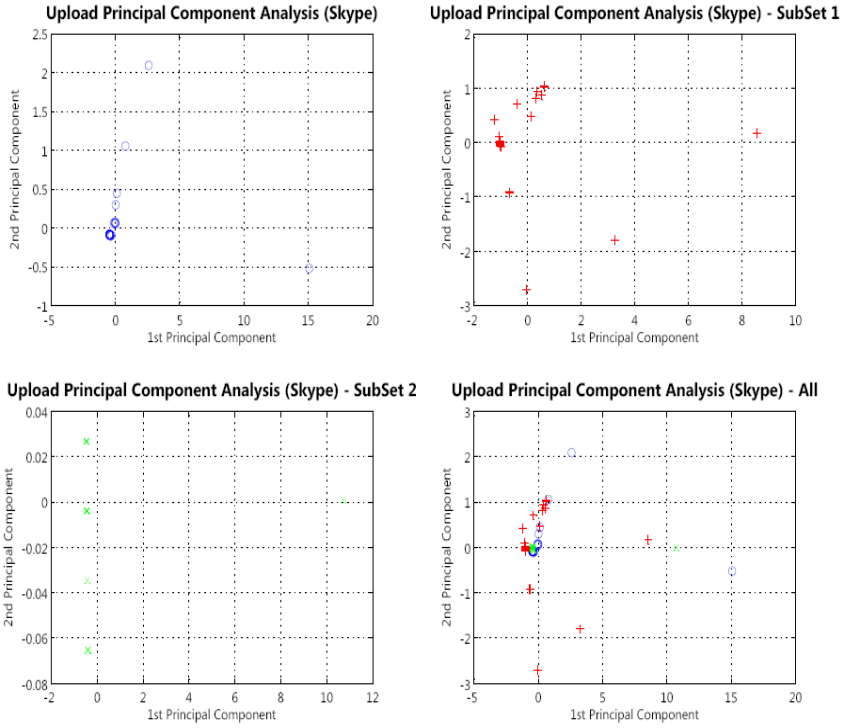

figura 54: *Skype Upload PCA*

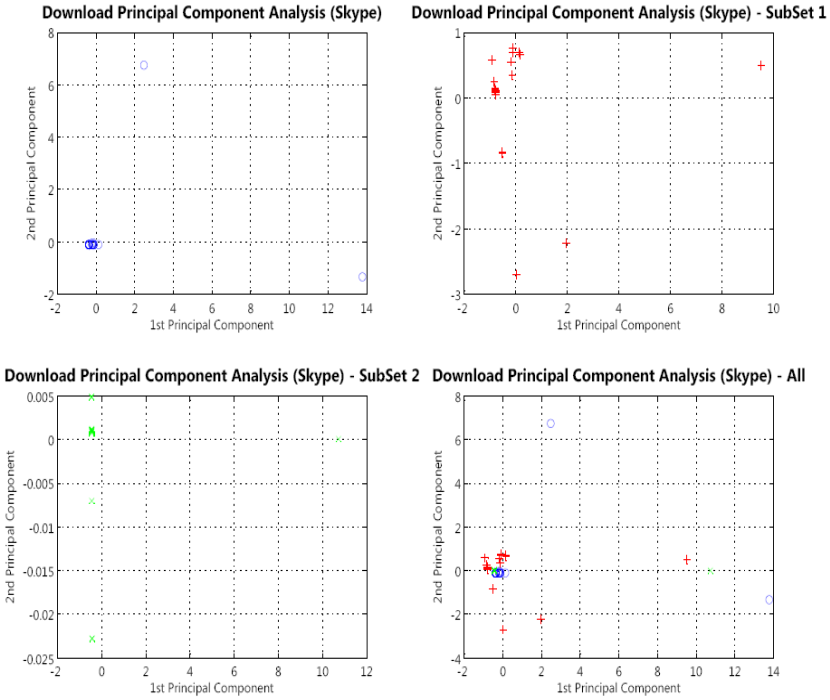

figura 55: *Skype Download PCA*

YOUTUBE PCA

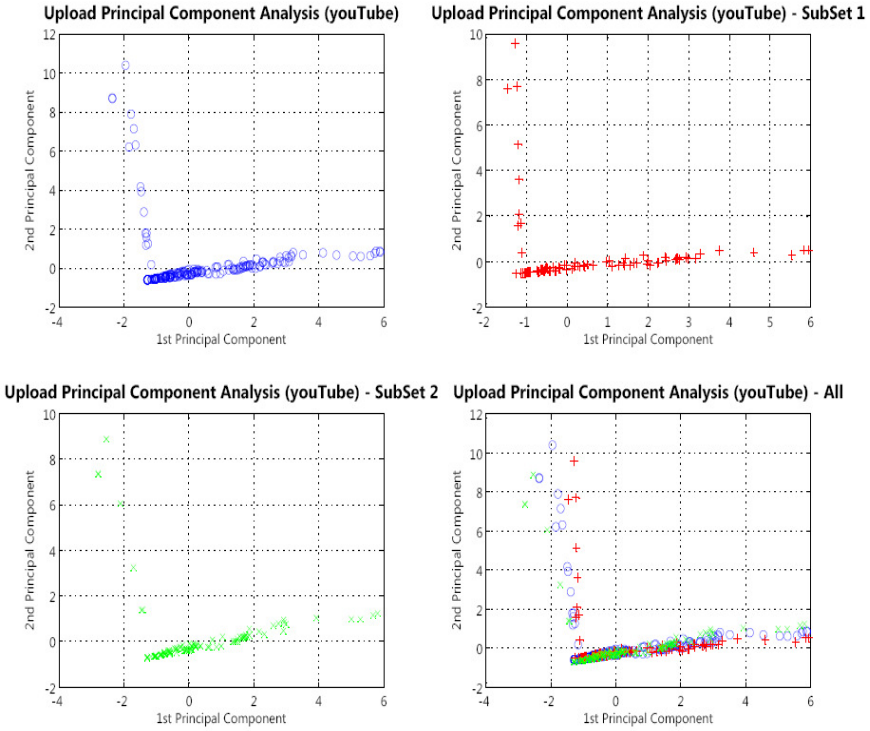

figura 56: *youTube Upload PCA*

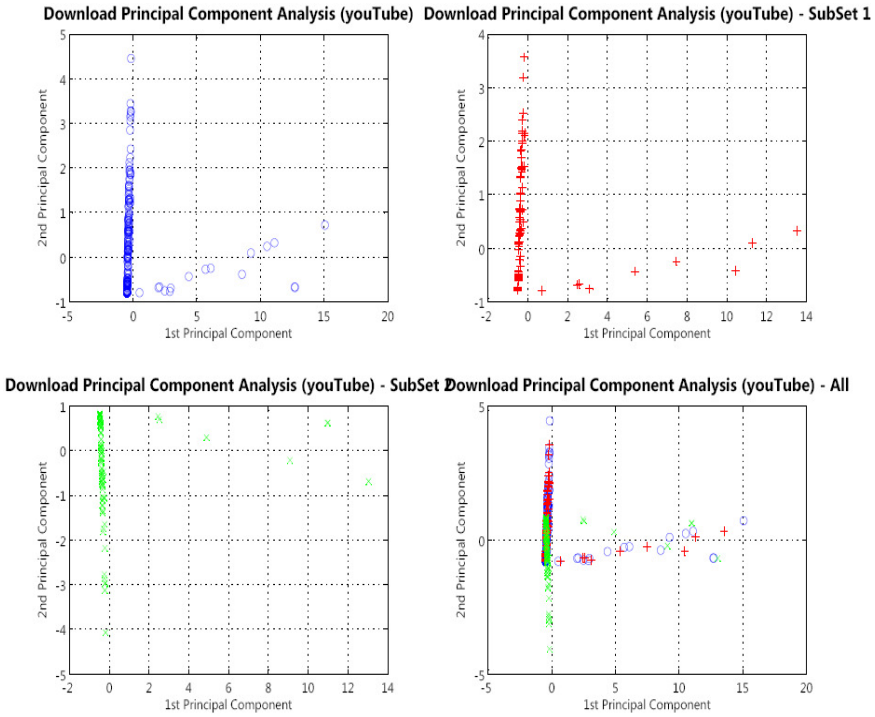

figura 57: *youTube Download PCA*

BIBLIOGRAFIA

[1] Karl Aberer, Manfred Hauswirth: "*An Overview on Peer-to-Peer Information Systems*". Swiss Federal Institute of Technology (EPFL), Switzerland, 2002.

[2] Wayne Blackard: "*Managing Peer-to-Peer Traffic In Network Environments*".

[3] José Rodrigo Furlanetto de Azambuja: "*Peer to Peer: Modelos, Middlewares e Aplicações*".

[4] João Rocha, Marco Domingues, Arthur Callado, Eduardo Souto, Guthemberg Silvestre, Carlos Kamienski, Djamel Sadok: "*Peer-to-Peer: Computação Colaborativa na Internet*".

[5] Cachelogic A. Parker: "*The true Picture of peer-to-peer file sharing*". http://www.cachelogic.com/research/slide9.php.

[6] Dario Rossi PhD Thesis: "*Traffic at the Network Edge:Sniff, Analyze, Act*". Politecnico di Torino, 2005.

[7] "*TSTAT – TCP Statistic and Analysis Tool*". http://tstat.tlc.polito.it

[8] Karen Kent Frederick: "*Studying Normal Network Traffic*", 2001.

[9] "*WinDump – Tcpdump for Windows*". http://www.winpcap.org/windump

[10] Jonathon Shlens: "*A Tutorial on Principal Component Analysis*",2005.

[11] Daniela Raicu: "*Advanced Statistics with MATLAB*". DePaul University, Chicago, 2004.

[12] "*Bioinformatics Toolbox – Principal Component Analysis*". http://www.mathworks.com/access/helpdesk_r13/help/toolbox/bioinfo/a1060813273b1.html

[13] Brian Kantor, Phil Lapsley: "*RFC977 – Network News Transfer Protocol - A Proposed Standard for the Stream-Based Transmission of News*", 1986.

[14] Jonathan B. Postel: "*RFC821 – Simple Mail Transfer Protocol*", 1982.

[15] Salman A. Baset and Henning G. Schulzrinne: "*An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol*". Columbia University, New York.

[16] "*Skype – Talk with anyone, anywhere in the world*". http://www.skype.com

[17] "*Napster website*". http://www.napster.com

[18] "*OpenNap – Open Source Napster Server*". http://opennap.sourceforge.net

[19] T. Klingberg, R. Manfredi: "*Gnutella 0.6*". Network Working Group, 2002.

[20] Marcus Bergner: "*Improving performance of modern Peer-to-Peer services*". Umea University, 2003.

[21] "*The giFT Project: Internet File Transfer*". http://gift.sourceforge.net

[22] "*Kazaa: A completely distributed peer-to-peer file sharing service*". http://www.kazaa.com

[23] Yoram Kulbak, Danny Bickson: "*The eMule Protocol Specification*". The Hebrew University of Jerusalem, Jerusalem, 2005.

[24] Bram Cohen: "*BitTorrent Protocol Specification*". http://www.bittorrent.org/protocol.html

[25] Aaron Harwood, Thomas Jacobs: "*Localhost: A browsable peer-to-peer file sharing system*", 2005.

[26] K. Raeburn: "*Unkeyed SHA-1 Checksum Specification*". IETF, 2004.

[27] John G. Waclawsky: "*P2P: The Next Wave Of Internet Evolution*". Business Communications Review, 2006.

[28] "Shareaza: Multi-network peer-to-peer (*P2P*) file-sharing client". http://sourceforge.net/projects/shareaza

[29] "*CVS – Concurrent Version Systems*". http://www.cvshome.org

[30] "*7 things you should know about... youTube*", 2006. http://www.educause.edu/eli

[31] "*youTube – Broadcast YourSelf*". http://www.youtube.com

[32] The Math Works Inc.: "*Getting Started with MATLAB, Version 6*", 2001.

[33] The Math Works Inc.: "*Using MATLAB Graphics, Version 5.3*", 1999.

[34] The Math Works Inc.: "*Statistics Toolbox for use with MATLAB, Version 2*", 1999.